SILVACO International

ATLAS User's Manual

DEVICE SIMULATION SOFTWARE

Volume II

SILVACO International
4701 Patrick Henry Drive, Bldg. 1
Santa Clara, CA 95054
Telephone (408) 567-1000
FAX: (408) 496-6080

February 2000

ATLAS User's Manual

Copyright 2000
SILVACO International
4701 Patrick Henry Drive, Building 1
Santa Clara, CA 95054

Phone: (408) 567-1000
FAX: (408) 496-6080

Notice

The information contained in this document is subject to change without notice.

SILVACO International MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE.

SILVACO International Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

This document contains proprietary information, which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated into another language without the prior written consent of **SILVACO** International.

Simulation Standard, TCADDrivenCAD, Virtual Wafer Fab, Analog Alliance, Legacy, ATHENA, ATLAS, FASTATLAS, ODIN, VYPER, CRUSADE, RESILIENCE, DISCOVERY, CELEBRITY, Production Tools, Automation Tools, Interactive Tools, TonyPlot, DeckBuild, DevEdit, Interpreter, ATHENA Interpreter, ATLAS Interpreter, Circuit Optimizer, MaskViews, PSTATS, SSuprem3, SSuprem4, Elite, Optolith, Flash, Silicides, SPDB, CMP, MC Deposit, MC Implant, Process Adaptive Meshing, S-Pisces, Blaze, Device3D, Thermal3D, Interconnect3D, Blaze 3D, Giga3D, MixedMode3D, TFT, Luminous, Giga, MixedMode, ESD, Laser, FastBlaze, FastMixedMode, FastGiga, FastNoise, MOCASIM, UTMOST, UTMOST II, UTMOST III, UTMOST IV, PROMOST, SPAYN, SmartSpice, MixSim, Twister, FastSpice, SmartLib, SDDL, EXACT, CLEVER, STELLAR, HIPEX, Scholar, SIREN, Escort, Starlet, Expert, Savage, Scout, Guardian, and Envoy – are trademarks of **SILVACO** International.

© 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 2000 by **SILVACO** International Inc.

Reader Comment Sheet

We welcome your evaluation of this manual. Your comments and suggestions help us to improve our publications. If you have any responses to the questions below, please let us know. Write your observations, complaints, bug reports, suggestions or comments below.

- Is this manual technically accurate?
- Are the concepts and wording easy to understand?
- Is the size of this manual convenient for you?
- Is the manual's arrangement convenient for you?
- Do you consider this manual to be easily readable?
- Please add any additional relevant comments.

Please fax your comments to **SILVACO** International, Attention Technical Publications at (408) 496-6080.

Introduction

Intended Audience

The information presented is based on the assumptions that the reader is (1) familiar with the basic terminology of semiconductor processing and semiconductor device operation, and (2) understands basic operation of the computer hardware and operation system being employed.

Introduction

ATLAS is a modular and extensible framework for one, two and three dimensional semiconductor device simulation. It is implemented using modern software engineering practices that promote reliability, maintainability, and extensibility. Products that use the ATLAS Framework meet the device simulation needs of all semiconductor application areas.

SILVACO International makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranty of fitness for a particular purpose.

SILVACO International shall not be liable for errors contained herein or for incidental or consequential damages in connection with furnishing, performance, or use of this material. This document contains proprietary information protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated into another language without the prior written consent of **SILVACO** International.

Editions are recorded below under History, and are individually listed as Edition 1 through 6. The basic issue of the manual is Edition 1. The date is also noted. A completely revised manual results in a new edition.

History

- Edition 1 - July 1, 1993
- Edition 2 - March 1, 1994
- Edition 3 - June 1, 1994
- Edition 4 - October 30, 1996
- Edition 5 - April 30, 1997
- Edition 6 - November 1998
- Edition 7 - February 2000

This page intentionally left blank.

Table of Contents

Chapter 11: 3D Device Simulation 11-1

Overview of 3D Device Simulation Programs 11-1
- Device3D 11-1
- Blaze3D 11-1
- Giga3D 11-1
- TFT3D 11-2
- MIXEDMODE3D 11-2
- Quantum3D 11-2
- Luminous 11-2

3D Structure Generation 11-3
- ATLAS Syntax For 3D Structure Generation 11-3
- DevEdit3D Interface 11-3

Model And Material Parameter Selection in 3D 11-4
- Simulation of Single Event Upset 11-5
- User-defined SEU in 3D 11-7

Boundary Conditions in 3D 11-7
- External Passive Elements 11-7
- Thermal Contacts for GIGA3D 11-7

BLAZE3D Models 11-8
TFT3D Models 11-8
QUANTUM3D Models 11-8
LUMINOUS3d Models 11-8

Numerical Methods for 3D 11-12
- DC Solutions 11-12
- Transient Solutions 11-12
- Obtaining Solutions In 3D 11-12
- Interpreting the Results From 3D 11-12
- More Information 11-12

Chapter 12: INTERCONNECT3D 12-1

Overview 12-1

Model Description 12-1

Capacitance and Conductance Calculation 12-1
- Boundary Conditions 12-1

Structure Definition 12-3

Tutorial For The MaskViews Interface 12-4
- Saving Mask Data For INTERCONNECT3D 12-4
- Loading Mask Data Into INTERCONNECT3D 12-4

Defining Electrodes 12-4

Model And Material Parameter Selection 12-6

Conductivity Calculations 12-6

Capacitance Calculations 12-6
Applying CD Variations and Misalignment 12-7
Numerical Methods 12-7
Obtaining Solutions 12-7
Interpreting the Results 12-7
More Information 12-7

Chapter 13: THERMAL3D 13-1

Overview 13-1
3D Structure Generation 13-1
Defining Heat Sources 13-1
Defining Heat Sinks 13-1
Model and Material Parameter Selection 13-2
Setting Thermal Conductivity 13-2
Suggested Parameters For Thermal Conductivity 13-3
Numerical Methods 13-3
Obtaining Solutions In THERMAL3D 13-3
Interpreting The Results From THERMAL3D 13-4
More Information 13-4

Chapter 14: Numerical Techniques 14-1

Overview 14-1
Numerical Solution Procedures 14-1
Meshes 14-1
Mesh Regridding 14-2
Mesh Smoothing 14-3
Discretization 14-3
The Discretization Process 14-3
Non-Linear Iteration 14-4
Newton Iteration 14-4
Gummel Iteration 14-4
Block Iteration 14-5
Combining The Iteration Methods 14-5
Solving Linear Subproblems 14-5
Convergence Criteria for Non-linear Iterations 14-6
Error Measures 14-6
Carrier Concentrations and CLIM.DD (CLIMIT) 14-6
Discussion of CLIM.EB 14-7
Terminal Current Criteria 14-7
Summary of Termination Criteria 14-8
Detailed Convergence Criteria 14-10
Convergence Criteria For Gummel's Algorithms 14-10
Convergence Criteria For Newton's Algorithm 14-12
Convergence Criteria For Block Iteration 14-14

Initial Guess Strategies 14-15
Recommendations And Defaults 14-16
The DC Curve-Tracer Algorithm 14-16
Transient Simulation 14-17
Small Signal and Large Signal Analysis 14-18
Frequency Domain Perturbation Analysis 14-18
Fourier Analysis Of Transient Responses 14-19
Overall Recommendations 14-20
Differences Between 2D and 3D Numerics 4-21

Chapter 15: Statements 15-1

Overview 15-1
Input Language 15-1
Syntax Rules 15-1
Mnemonics 15-2
Continuation Lines 15-2
Comments 15-2
Synonyms 15-2
Pseudonyms 15-2
Symbols 15-3
Expressions. 15-3
BEAM 15-4
Syntax 15-4
Description 15-6
Monochromatic Beam Example 15-8
Multispectral Beam Example 15-8
COMMENT, # 15-9
Syntax 15-9
Example 15-9
CONTACT 15-10
Syntax 15-10
Description 15-11
Workfunction Parameters 15-11
Boundary Conditions 15-12
Contact Parasitics 15-13
Electrode Linking Parameters 5-13
Floating Gate Capacitance Parameters 15-13
Schottky Barrier and Surface Recombination Example 15-13
Parasitic Resistance Example 15-14
Floating Gate Example 15-14
CURVETRACE 15-15
Syntax 15-15
Description 15-15
Diode Breakdown Example 15-16
DEFECT 15-17
Syntax 15-17

Description 15-18
TFT Example 15-19
DEGRADATION 15-20
Syntax 15-20
Description 15-20
MOS Interface State Example 15-20
DOPING 15-21
Syntax 15-21
Description 15-23
Analytical Profile Types 15-23
File Import Profile Types 15-24
Parameters that Specify the Dopant Type 15-24
Vertical Distribution Parameters 15-25
Location Parameters 15-25
Lateral Extent Parameters 15-26
Lateral Distribution Parameters 15-26
Trap Parameters 15-26
ELECTRODE 15-29
Syntax 15-29
Description 15-29
Position Parameters 15-30
Region Parameters 15-29
MOS Electrode Definition Example 15-31
3D Electrode Definition Example 15-31
ELIMINATE 15-32
Substrate Mesh Reduction Example 15-33
EXTRACT 15-34
Terminal Current Extraction Example 15-34
Extraction Example from Previously Generated Results 15-34
Solution Quantities Extraction Example 15-34
FOURIER 15-35
Example 1: 15-36
Example 2: 15-36
GO 15-37
Example starting a given ATLAS Version 15-37
Parallel ATLAS Example 15-37
IMPACT 15-38
Description 15-39
Model Selection Flags 15-39
Model Localization Parameters 15-40
Selberrherr Model Parameters 15-40
Temperature Dependence Parameters 15-40
Parameters for use with Energy Balance 15-40
Concannon Model Parameters 15-41
Selberrherr Model Example 15-41

INTERFACE 15-42
Description 15-42
Boundary Condition Parameters 15-42
Position Parameters 15-43
MOS Example 15-43
SOI Example 15-43
Interface Charge for III-V Devices 15-43
INTTRAP 15-44
Description 15-44
Capture Parameters 15-45
Example setting Multiple Interface Trap States 15-45
LOAD 15-46
Description 15-46
File Parameters 15-46
Simple Save and Load Examples 15-47
Binary Format Example 1 5-47
LOG 15-48
File Output Parameters 15-49
RF Analysis Parameters 15-49
Parasitic Element Parameters 15-50
Simple Logfile Defintion Example 15-50
RF Analysis Example 15-50
Transient or AC Logfile Example 15-50
LX.MESH, LY.MESH 15-52
Description 15-52
LASER Mesh Example 15-52
MATERIAL 15-53
Description 15-57
Localization of Material Parameters 15-57
Band Structure Parameters 15-57
Mobility Model Parameters 15-58
Recombination Model Parameters 15-58
Carrier Statistics Model Parameters 15-60
Energy Balance Parameters 15-60
Lattice Temperature Dependence Parameters 15-60
Oxide Material Parameters 15-61
Photogeneration Parameters 15-61
LASER Parameters 15-61
Material Coefficient Definition Examples 15-62
MEASURE 15-63
Description 15-64
Data Type Parameters 15-64
Boundary Parameters 15-65
Resistance Example 15-65
Gate Charge Example 15-65
Ionization Integral Example 15-65
MESH 15-66
Description 15-66

Parameters related to reading in an existing mesh file 15-66
Parameters Related to Creation of a New Mesh 15-67
Output Parameters 15-67
Mesh Definition Example 15-67
ATHENA Interface Example 15-68
METHOD 15-69
Description 15-71
Parameters to select the Solution Method 15-71
Parameters to select which equations are solved 15-72
Solution Tolerance Parameters 15-72
General Parameters 15-73
Gummel Parameters 15-74
Newton Parameters 15-74
Numerical Method Defintion Example 15-74
TRAP Parameter Example 15-75
Transient Method Example 15-75
MOBILITY 15-76
Description 15-84
Mobility Model Flags 15-84
Example Selecting the Modifed Watt Model 15-91
MODELS 15-92
Description 15-97
Mobility Model Flags 15-98
Recombination Model Flags 15-98
Generation Model Flags 15-99
Classical Carrier Statistics Model Flags 15-99
Quantum Carrier Statistics Model Flags 15-100
Energy Balance Simulation Flags 15-100
Lattice Heating Simulation Flags 15-101
Model Macros 15-101
General Parameters 15-101
Model Dependent Parameters 15-101
Model Selection Example 15-104
Confirming Model Selection 15-104
OPTIONS 15-105
Description 15-105
Example 15-105
OUTPUT 15-106
Description 15-108
Ionization Integral Parameters 15-110
Averaging Parameters for Vector Quantities 15-110
Example of combining OUTPUT with SOLVE and SAVE 15-110
PROBE 15-111
Description 15-112
Example of Probing the Maximum Value 15-113
Example of PROBE at a location 15-113
Vector Quantitiy Example 15-113
QUIT 15-114

Description 15-114
REGION 15-115
Description 15-115
Position Parameters 15-116
Grid Inidices Example 15-117
Non-Rectangular Region Example 15-117
Typical MOS Example 15-117
3D Region Definition Example 15-117
REGRID 15-118
Description 15-119
Variable Parameters 15-119
Location Parameters 15-119
Control Parameters 15-120
File I/O Parameters 15-120
Doping Regrid Examples 15-121
Potential Regrid Example 15-121
Re-initializing after regrid example 15-121
SET 15-122
Numeric Variable Example 15-122
String Variable Example 15-122
SAVE 15-123
Description 15-123
Basic Save Example 15-123
Example with User-defined Output 15-123
SINGLEEVENTUPSET 15-125
Description 15-125
SEU Example 15-126
SOLVE 15-127
Description 15-130
DC Parameters 15-131
File Output Parameters 15-132
Initial Guess Parameters 15-132
Compliance Parameters 15-132
Transient Parameters 15-133
AC Parameters 15-134
Ionization Integral Parameters 15-134
Photogeneration Parameters 15-135
Thermal3D parameters 15-136
DC Conditions Example 15-136
Bias Stepping Example 15-136
Transient Simulation Example 15-136
AC Analysis Example 15-137
Photogeneration Examples 15-137
Ionization Integral Example 15-138
SPREAD 15-139
Description 15-139
Mandatory Parameters 15-139
Examples 15-139

SYMBOLIC 15-142
SYSTEM 15-143
Examples 15-143
THERMCONTACT 15-144
Description 15-144
Position Parameters 15-144
Coordinate Definition Example 15-145
TONYPLOT 15-146
Examples 15-146
TITLE 15-147
Example 15-147
TRAP 15-148
Description 15-148
Capture Parameters 15-149
Multiple Trap Level Definition Example 15-149
UTMOST 15-150
Description 15-151
X.MESH, Y.MESH, Z.MESH 15-154
Description 15-154
Example Setting Fine Grid at A Junction 15-154

Appendix A:
C-Interpreter Functions A-1

Introduction A-1

Appendix B:
Material Systems B-1

Overview B-1
Semiconductors, Insulators and Conductors B-1
Semiconductors B-1
Insulators B-1
Conductors B-1
Unknown Materials B-2
ATLAS Materials B-3
Rules for Specifying Compound Semiconductors B-4
Silicon and Polysilicon B-6
Silicon and Polysilicon Band Parameters B-6
Silicon and Polysilicon Dielectric Properties B-6
Silicon and Polysilicon Default Mobility Parameters B-6
Silicon and Polysilicon Bandgap Narrowing Parameters B-7
Silicon and Polysilicon Recombination Parameters B-7
Silicon and Polysilicon Impact Ionization Coefficients B-8
Silicon and Polysilicon Thermal Parameters B-8
Silicon and Polysilicon Effective Richardson Coefficients B-8
The Al(x)Ga(1-x)As Material System B-10
A1GaAs Recombination Parameters B-10
GaAs and A1GaAs Impact Ionization Coefficients B-10

A1GaAs Thermal Parameters **B-11**
GaAs Effective Richardson Coefficients **B-11**
The In(1-x)Ga(x)As(y)P(1-y) System **B-12**
Silicon Carbide (SiC) **B-14**
SiC Thermal Parameters B-14
Miscellaneous Semiconductors **B-15**
Miscellaneous Semiconductor Band Parameters B-15
Miscellaneous Semiconductor Dielectric Properties B-16
Miscellaneous Semiconductor Mobility Properties B-17
Insulators **B-19**
Insulator Dielectric Constants B-19
Insulator Thermal Properties B-17
Optical Properties **B-19**
User Defined Materials **B-22**

Appendix C:
Hints and Tips **C-1**

Appendix D:
ATLAS Version History **D-1**

Bibliography

Volume 1

Volume 2

Index

Volume 1

Volume 2

This page intentionally left blank.

List of Figures

Figure No.	Caption Title	Page No.
11-1:	Source beam coordinate rotation around Z-axis	11-9
11-2:	Source beam coordinate rotation around Y-axis	11-9
11-3:	Source beam sampling	11-10
11-4:	LUMINOUS3D Lenslet Specification 1	1-11
12-1	The structure to be simulated	12-2
12-2	What INTERCONNECT3D actually simulates	12-2
12-3	Information flow for parasitic extraction from IC layout	12-3
14-1:	Load the algorithm used in the Curve Tracer	14-17
15-1	LUMINOUS Optical Source Coordinate System	15-5
C-1	Simple ray trace in LUMINOUS	C-2
C-2	Addition of back and sidewall reflection to Figure C-1	C-2
C-3	Photogeneration contours based on ray trace in Figure C-2	C-3
C-4	Comparison of CPU time showing advantages of decoupled methods at low current and coupled methods at high currents	C-4
C-5	High frequency CV curve showing poly depletion effects at positive Vgs.	C-6
C-6	Figure 2 Electron concentration profile of an NMOS transistor poly depletion occurs at the poly/gate oxide interface.	C-6
C-7	Fine Grid - 5000 nodes	C-8
C-8	Fine Grid only in Key Areas - 2450 nodes	C-8
C-9	Id-Vds Curves for two MOSFET Meshes	C-9
C-10	Breakdown Voltage for Two MOSFET Meshes	C-9
C-11	Electric field in MOS gate oxide during a high current pulse on the drain.	C-11
C-12	Mobility (normalized) rolls off as a high gate electric field is applied	C-11
C-13	Using a PROBE of electron concentration allows a study of MOS width effect using 2D simulation. An enhanced electron concentration is seen along slice 2.	C-12
C-14	3D device simulation of MOS width effect can be performed on structures created ATHENA.	C-13

[This page intentionally left blank]

List of Tables

Table No.	Caption Title	Page No.
11-1	User Specifiable Parameters for Equations 11-3 and 11-4	11-5
11-2	User Specifiable Parameters for Equations 11-1, 11-2, 11-7 and 11-8	11-7
13-1	User Specifiable Parameters For Equations 13-2 to 13-5	13-2
14-1	User Specifiable Parameters for Convergence Criteria	14-9
14-2	User Specifiable Parameters for Equation 9-2	14-19
15-1	Types of Parameters	15-2
A-1	Complete list of available C-Interpreter functions in ATLAS	A-3
B-1	The ATLAS Materials	B-3
B-2	Band parameters for Silicon and Poly	B-6
B-3	Static dielectric constants for Silicon and Poly	B-6
B-4	Lattice Mobility Model Defaults for Silicon and Poly	B-6
B-5	Parallel Field Dependent Mobility Model Parameters for Silicon and Poly	B-6
B-6	Bandgap Narrowing Parameters for Silicon and Poly	B-7
B-7	SRH Lifetime Parameter Defaults for Silicon and Poly	B-7
B-8	Auger Coefficient Defaults for Silicon and Poly	B-7
B-9	Impact Ionization Coefficients for Silicon and Poly	B-8
B-10	Effective Richardson Coefficients for Silicon and Poly	B-8
B-11	Effective Richardson Coefficients for Silicon and Poly	B-9
B-12	Default Recombination Parameters for AlGaAs	B-10
B-13	Impact Ionization Coefficients for GaAs	B-10
B-14	Default Thermal Parameters for GaAs	B-11
B-15	Thermal Resistivities for InGaAsP Lattice-Matched to InP	B-12
B-16	Default Thermal Properties of InP InAs GaP and GaAs	B-12
B-17	Impact Ionization Coefficients for SiC	B-14
B-18	Default Thermal Parameters for SiC	B-14
B-19	Band Parameters for Miscellaneous Semiconductors	B-15
B-20	Static Dielectric Constants for Miscellaneous Semiconductors	B-16
B-21	Mobility Parameters for Miscellaneous Semiconductors	B-17
B-22	Default Static Dielectric Constants of Insulators	B-19
B-23	Default Thermal Parameters for Insulators	B-19
B-24	Wavelength Ranges for Default Complex Index of Refraction	B-21

[This page intentionally left blank]

Chapter 11:
3D Device Simulation

Overview of 3D Device Simulation Programs

Note: This chapter aims to highlight the extra information required for 3-D simulation as compared to 2-D. Users should be familiar with the equivalent 2-D models before reading this chapter.

This chapter describes the set of ATLAS products that extends 2D simulation models and techniques and applies them to general non-planar 3D structures. The structural definition, models and material parameters settings and solution techniques are similar to 2D. Users should be familiar with the simulation techniques described in the 'Getting Started' and equivalent 2D product chapters before reading the sections that follow. The products that form 3D device simulation in ATLAS are:

- DEVICE3D – silicon simulation equivalent to S-PISCES
- BLAZE3D – compound material and heterojunction simulation
- GIGA3D – non-isothermal simulation
- MIXEDMODE3D – mixed device-circuit simulation
- TFT3D – thin film transistor simulation
- QUANTUM3D – quantum effects simulation
- LUMINOUS3D – photodetection simulation

Two other 3D modules, INTERCONNECT3D and THERMAL3D are documented in separate chapters.

In a similar manner to the 2D products, GIGA3D, MIXEDMODE3D, LUMINOUS3D and QUANTUM3D should be combined with both DEVICE3D or BLAZE3D depending on the semiconductor materials used.

DEVICE3D

Device3D provides semiconductor device simulation of silicon technologies. It's use is anagous to the 2-D simulations in S-PISCES. See Chapter 4 for more information on particular simulation techniques.

BLAZE3D

BLAZE3D allows simulation of semiconductor devices with semiconductor compositional variations (i.e., heterojunction devices). Blaze3D is completely analogous to Blaze (see Chapter 5), with some exceptions. First, the current version of Blaze3D does not allow or account for compositional variations in the Z direction. Second, the current version does not account to thermionic interfaces at abrupt heterojunctions .

GIGA3D

Giga3D is an extension of Device3D or Blaze3D that accounts for lattice heat flow in 3D devices. Giga3D has all the functionality of GIGA (see Chapter 6) with a few exceptions. First, additional syntax has been added to account for the three dimensional nature of thermal contacts. The parameters `Z.MIN` and `Z.MAX` can be specified on the `THERMCONTACT` statement to describe the extent of the contact in the Z direction. The other exception in the current version is that there is no `BLOCK` method available in the 3D version.

TFT3D

TFT3D is an extension of DEVICE3D that allows the simulation of amorphous and polycrystalline semiconductor materials in three dimenstions. TFT3D is completely analogous to the TFT simulator described in Chapter 7. The complete functionality of the TFT simulator is available in TFT3D for three dimensional devices.

MIXEDMODE3D

MIXEDMODE3D is an extension of DEVICE3D or BLAZE3D that allows the simulation of physical devices embedded in lumped element circuits (Spice circuits). MIXEDMODE3D is completely analogous to the MIXEDMODE simulator described in the Chapter 10. The complete functionality of the MIXEDMODE simulator is available in MIXEDMODE3D for three dimensional devices.

QUANTUM3D

QUANTUM3D is an extension of DEVICE3D or BLAZE3D that allows the simulation of the effects of quantum confinement using the quantum transport model (see Chapter 3). QUANTUM3D is completely analogous to the QUANTUM model, but applies to three dimensional devices.

LUMINOUS3D

LUMINOUS3D is an extension of DEVICE3D that allows the simulation of photodetection in three dimenstions. LUMINOUS3D is analogous to the LUMINOUS simulator described in Chapter 8 with a few significant differences described later in this chapter.

3D Structure Generation

All 3-D programs in ATLAS supports structures defined on 3D prismatic meshes. Structures may have arbitrary geometries in two dimensions and consist of multiple slices in the third dimension.

There are two methods for creating a 3D structure that can be used with ATLAS. One is through the command syntax of ATLAS and the other through an interface to DEVEDIT3D.

A direct interface from ATHENA to 3D ATLAS is not possible, however DEVEDIT3D provides the ability to read in 2D structures from ATHENA and extend them non-uniformly to create 3D structures for ATLAS.

ATLAS Syntax For 3D Structure Generation

Mesh generation

The 'Getting Started' chapter covers the generation of 2D and 3D mesh structures using the ATLAS command language. The Z.MESH statement and the NZ and THREE.D parameters of the MESH statement are required to extend a 2D mesh into 3D.

By convention, slices are made perpendicular to the Z axis. The mesh is triangular in XY but rectangular in XZ or YZ planes.

Region, Electrode and Doping definition

The Getting Started chapter covers the definition of 2D regions, electrodes and doping profiles. In order to extend the regions into 3D the Z.MIN and Z.MAX parameters are used. For example:

```
REGION NUM=2 MATERIAL=Silicon X.MIN=0 X.MAX=1 Y.MIN=0 Y.MAX=1 Z.MIN=0
Z.MAX=1
ELECTRODE NAME=gate X.MIN=0 X.MAX=1 Y.MIN=0 Y.MAX=1 Z.MIN=0 Z.MAX=1
DOPING GAUSS N.TYPE CONC=1E20 JUNC=0.2 Z.MIN=0.0 Z.MAX=1.0
```

For 2D regions or electrodes defined using the command language, geometry is limited to rectangular shapes. Similarly, in 3D regions and electrodes are composed of rectilinear parallelopipeds.

DEVEDIT3D Interface

DEVEDIT3D is a graphical tool that allows users to draw 3D device structures and create 3D meshes. It can also read 2D structures from ATHENA and extend them into 3D. These structures can be saved from DEVEDIT3D as structure files for use by ATLAS. When using DEVEDIT3D it is important to also save a command file. This file is used to recreate the 3D structure inside DEVEDIT3D and is important since DEVEDIT3D does not read in 3D structure files.

ATLAS can read in structures generated by DEVEDIT3D using the command

```
MESH INF=<filename>
```

The program is able to distinguish automatically between 2D and 3D meshes read in using this command.

Defining Devices with Circular Masks

DEVEDIT3D makes a triangular mesh in the XY plane and uses z-plane slices. This means that normally the Y direction is vertically down into the substrate. However in the case of using circular masks it is necessary to rotate the device.

For defining devices using circular masks in DEVEDIT3D the XY plane should be the surface of the device and the Z direction should be into the substrate.

Model And Material Parameter Selection in 3D

Models and material parameters are chosen in 3-D in common with other 2-D modules using the MODELS, IMPACT, MATERIAL, MOBILITY, INTERFACE, and CONTACT statements.

The models available in 3D device simulation programs are:

1.Mobility:

- Table for 300K (CONMOB)
- Thomas (ANALYTIC)
- Arora's Model (ARORA)
- Klaassen's Model (KLAASSEN)
- Lombardi's Model (CVT)
- Yamaguchi Model (YAMA)
- Parallel Field Dependence (FLDMOB)
- Parallel Field Dependence with negative differential mobility (FLDMOB EVSATMOD=1)

2. Recombination:

- Shockley Read Hall (SRH)
- Concentration dependent lifetime SRH (CONSRH)
- Klaassen's concentration dependent lifetime SRH (KLASRH)
- Auger (AUGER)
- Klaassen's concentration dependent Auger recombination model (KLAAUG)
- Optical Recombination (OPTR)
- Bulk and interface traps (TRAP, INTTRAP)
- Continuous defect states (DEFECT)

3. Generation:

- Selberherr Impact Ionization (IMPACT SELB)
- Crowell Impact Ionization (IMPACT CROWELL)
- Hot Electron Injection (HEI)
- Fowler Nordheim Tunneling (FNORD)
- Single Event Upset (SINGLEEVENTUPSET)

4. Carrier Statistics:

- Boltzmann (default)
- Fermi (FERMI)
- Band Gap Narrowing (BGN)
- Incomplete Ionization (INCOMPLETE)
- Quantum Mechanical Effects (QUANTUM)

5. Boundary Conditions:

- Ohmic and Schottky
- Current Boundary Conditions
- Lumped Element Boundary Conditions
- Distributed Contact Resistance

6. Interface and Bulk Traps (Discrete and Continuous):

7. Photogeneration with Ray Tracing (LUMINOUS3D):

All these models, with the exception of SINGLEEVENTUPSET, are documented in the Physics or 2D product chapters of this manual.

Simulation of Single Event Upset

The capability of single event upset/photogeneration transient simulation is included in 3D using the SINGLEEVENTUPSET statement. It allows specification of the radial, length and time dependence specification of generated charge along tracks. Each track is specified by Entry Point (x0,y0,z0), Exit Point (x1,y1,z1) and is assumed to be a cylinder with the radius r0.

The entry and exit points are specified by the ENTRYPOINT and EXITPOINT parameters of the SINGLEEVENTUPSET statement. These are character parameters that represent the ordered triplet coordinates of the entry and exit points of the particle track.

The electron/hole pairs generated at any point is a function of the radial distance r from the centre of the track to the point, the distance l along the track and the time t. This equation takes the form;

$$G(r,l,t) = (DENSITY*L1(l) + S*B.DENSITY*L2(l))*R(r)*T(t) \quad (11\text{-}1)$$

where DENSITY is the number of generated electron/hole pairs per cm^3 and the other terms are defined as follows.

If the user defined parameter PCUNITS is set on the SINGLEEVENTUPSET statement then B.DENSITY is the generated charge, in pC/μm, and the scaling factor S is;

$$S = \frac{1}{q\pi\ \mathbf{RADIUS}^2} \quad (11\text{-}2)$$

where RADIUS is a user defined parameter on the SINGLEEVENTUPSET statement as shown in Table 11.1.

If the parameter, PCUNITS, is **not** set then B.DENSITY is the number of generated electron/hole pairs in cm^{-3} and the scaling parameter S is unity.

The factor L1 and L2 are defined by the equations:

$$L1(l) = \mathbf{A1} + \mathbf{A2} \cdot l + \mathbf{A3}\exp(\mathbf{A4} \cdot l) \quad (11\text{-}3)$$

$$L2(l) = \mathbf{B1}(\mathbf{B2} + l \cdot \mathbf{B3})^{\mathbf{B4}} \quad (11\text{-}4)$$

where the parameters: A1, A2, A3, A4, B1, B2, B3 and B4 are user definable as shown in Table 11.1.

Table 11.1. User-Specifiable Parameters for Equations 11-3 and 11-4			
Statement	**Parameter**	**Default**	**Units**
SINGLEEVENTUPSET	A1	1	
SINGLEEVENTUPSET	A2	0	cm^{-1}
SINGLEEVENTUPSET	A3	0	
SINGLEEVENTUPSET	A4	0	cm^{-1}
SINGLEEVENTUPSET	B1	1	
SINGLEEVENTUPSET	B2	1	
SINGLEEVENTUPSET	B3	0	cm^{-1}
SINGLEEVENTUPSET	B4	0	

The factor R(r) is the radial parameter which is defined by one of two equations. The default is:

$$R(r) = \exp\left(-\frac{r}{\mathbf{RADIUS}}\right) \qquad 11\text{-}5$$

where r is the radial distance from the centre of the track to the point and RADIUS is a user definable parameter as shown in Table 11-2. An alternative expression may be chose if the RADIALGAUSS parameter is specified on the SINGLEEVENTUPSET statement. In this case R(r) is given by;

$$R(r) = \exp-\left(\frac{r}{\mathbf{RADIUS}}\right)^2 \qquad 11\text{-}6$$

The time dependency of the charge generation T(t) is controlled by the user, with the parameter, TC, through two functions;

For TC=0;

$$T(t) = \mathrm{deltafunction}(t - \mathbf{T0}) \qquad 11\text{-}7$$

For TC>0;

$$T(t) = \frac{2e^{-\left(\frac{t-\mathbf{TO}}{\mathbf{TC}}\right)^2}}{\mathbf{TC}\sqrt{\pi}\,\mathrm{erfc}(\frac{-\mathbf{TO}}{\mathbf{TC}})} \qquad 11\text{-}8$$

where TO and TC are parameters of the SINGLEEVENTUPSET statement.

Table 11.2. User-Specifiable Parameters for Equations 11-1, 11-2, 11-7 and 11-8			
Statement	**Parameter**	**Default**	**Units**
SINGLEEVENTUPSET	DENSITY	0.0	cm^{-3}
SINGLEEVENTUPSET	B.DENSITY	0.0	cm^{-3} or pC/μm
SINGLEEVENTUPSET	TO	0.0	s
SINGLEEVENTUPSET	TC	0.0	s
SINGLEEVENTUPSET	RADIUS	0.05	micron

User-defined SEU in 3D

In addition to the model described by the SINGLEEVENTUPSET statement, users can use the C-INTERPRETER to specify an arbitrary generation profile. This is specified using the syntax:

```
BEAM NUM=1 F3.RADIATE=myseu.c
.
SOLVE B1=1.0 RAMPTIME=1e-12 DT=1e-14 TSTOP=1e-7
```

The F3.RADIATE parameter indicates an external c-language subroutine conforming to the template supplied. The file, **myseu.c**, returns a time and position dependent value of carrier generation in 3-D. The value returned by **myseu.c**, is multiplied by the parameter B1 at each timestep.

Boundary Conditions in 3D

External Passive Elements

It is possible to attach external lumped resistors, capacitors and inductors to any contact. The syntax is the same as for the 2D products:

```
CONTACT NAME=drain RES=1e3 CAP=1e-12 L=1e-6
```

It is also possible to apply distributed resistances to contacts. The algorithm used for estimating contact area for 3D distributed contact resistance multiplies the contact perimeter in a given Z plane by the displacement in the Z direction. This algorithm will only work properly for planar contacts that do not vary in the Z direction. They may however abruptly terminate or start in the Z direction.

The units of lumped extrnal passive elements are ohms for resistors, Farads for capacitors and Henrys for inductors. Distributed contact resistance is defined in ohms.cm^3.

Thermal Contacts for GIGA3D

Thermal contacts for non-isothermal simulation in GIGA3D are defined in an analagous manner to the 2D thermal contacts in GIGA. The Z.MIN and Z.MAX parameters are used to define the extent of the thermal contact in the z-plane. The units of the thermal resistance parameter ALPHA are scaled in 3D to W/(cm.K)

BLAZE3D Models

Models for simulating III-V materials are supported in BLAZE3D. The material parameters for all compound semiconductors are the same as used in 2-D simulations with BLAZE. Specialized models for III-V materials including heterojunction transport, optical recombination and negative differential field dependent mobility are documented in Chapter 5.

TFT3D Models

Models for simulating thin-film transistors made from amorphous or polycrystalline semiconductors are supported in TFT3D. The definition of the continuous defect states in the bandgap is performed using the same parameters as in 2D simulations with TFT. The models for continuous defect (or trap) densities are documented in Chapter 7.

QUANTUM3D Models

Models for simulating quantum effects semiconductors are supported in QUANTUM3D. The definition of the quantum moments solver is the same as in 2D simulations with QUANTUM. The models for simulating quantum effects and the parameters to control the model are documented in QUANTUM Models section of Chapter 3.

LUMINOUS3D Models

Many of the models for simulating photodetection in LUMINOUS3D are similar to those for simulating photodetection in LUMINOUS thus it is useful for the user to first read Chapter 8 regarding the use of LUMINOUS. There are. however, several important differences between LUMINOUS and LUMINOUS3D as described below.

Optical source specification

The specification of the optical source in 3D is similar to the specification of the source described in Chapter 8, LUMINOUS, with a few additions to account for the third dimension. Referring to Figures 11-1 and 11-2, the complete specification of the optical source in 3D includes the specification of the three coordinates of the source origin using the X.ORIGIN, Y.ORIGIN and Z.ORIGIN parameters of the BEAM statement as well as two angles of rotation using the THETA and PHI parameters. The parameter PHI is analogous to the LUMINOUS parameter ANGLE and specifies the direction of propogation relative to the device x axis (see Figure 11-1). In face PHI and ANGLE are synonomous. The parameter THETA specifies the rotated angle of propogation relative to the x-y plane (see Figure 11-2).

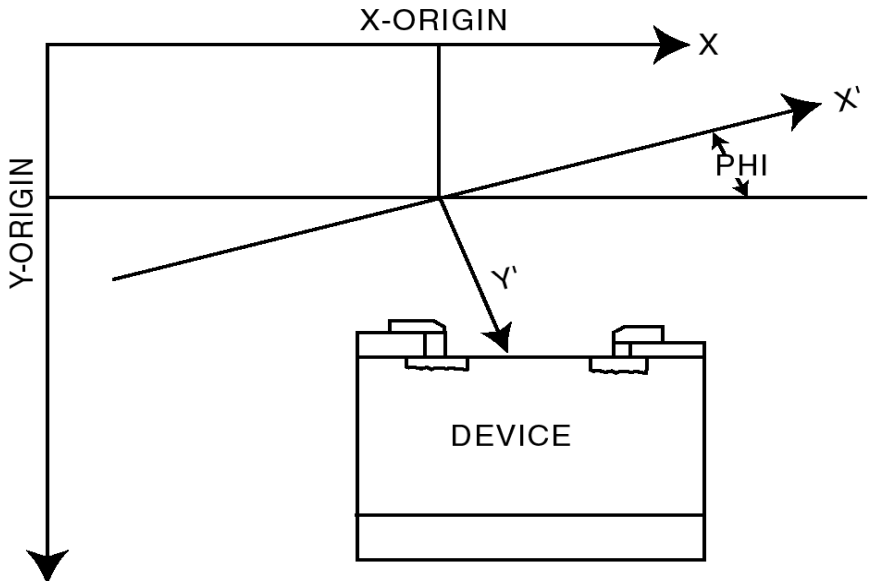


Figure 11-1: Source beam coordinate rotation around Z-axis

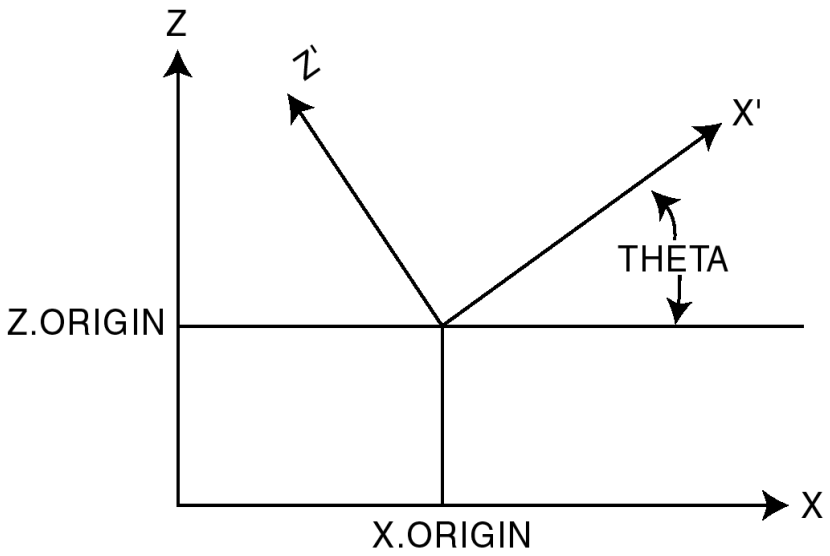


Figure 11-2: Source beam coordinate rotation around Y-axis

Ray tracing

The discrete sampling of the source beam into rays in Luminous3D is unlike that done in Luminous. In Luminous the source beam is automatically broken up into a set of "rays" that resolve the device topology and variations in the interior of the device. In 3D this process is much more complex and introduces an excessive computational burden. As such, in LUMINOUS3D the user must specify a discrete sampling of the source beam. This sampling is described in Figure 11-3.

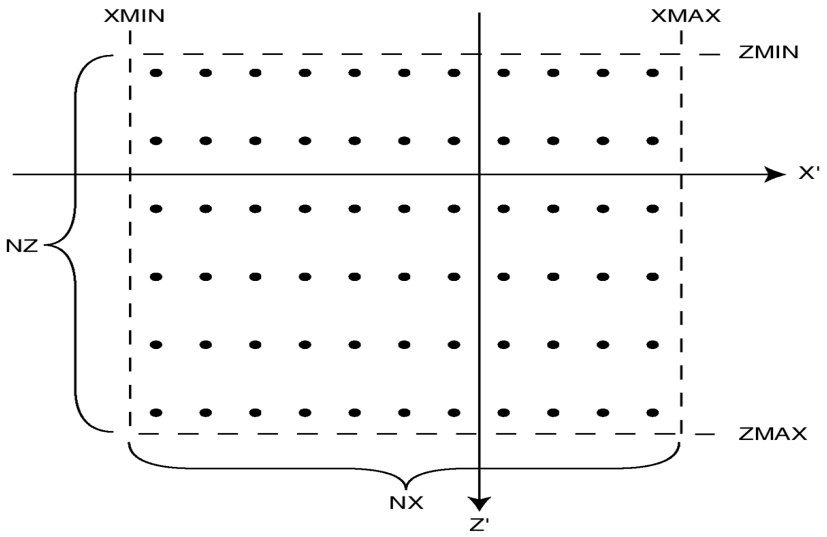


Figure 11-3: Source beam sampling

In Figure 11-3, the extent of the source sampling is specified by the XMIN, XMAX, ZMIN and ZMAX parameters of the BEAM statement. Even samples are taken along each of the beam front principal axes. The number of samples in the x' and z' directions are given by the NX and NZ parameters of the BEAM statement.

Lenslet specification

Another significant difference between LUMINOUS and LUMINOUS3D is that in 3-D the user may specify a "virtual" lenslet. This capability is provided as a convenience. Due to the restriction of prismatic elements in 3D, the user cannot accurately specify focusing elements using the device mesh.

In Luminous3D the user can specify a "virtual" lenslet as illustrated in Figure 11-4.

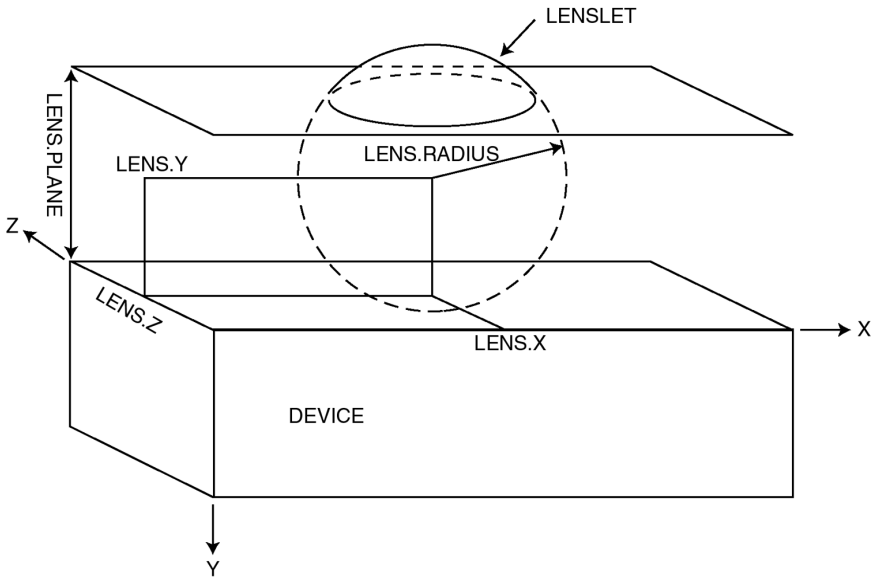


Figure 11-4: LUMINOUS3D Lenslet Specification

As shown in Figure 11-4, the lenslet is represented by a spherical section atop a plane. The center of the sphere is specified by the LENS.X, LENS.Y and LENS.Z parameters of the BEAM statement. The radius of the sphere is specified by the LENS.RADIUS parameter of the BEAM statement. The location of the planar part surrounding the lenslet relative to the y=0 plane is specified by the LENS.PLANE parameter. The index of refraction of the lenslet as well as the volume between the lens plane and the device surface is specified by the LENS.INDEX parameter of the BEAM statement. It should be noted that the volume associated with the lenslet is not meshed or considered in the solution of the device equations. It is meerly used in the ray tracing. Also the user should keep in mind that lenslet can only be specified in planes perpendicular to the y axis.

Other differences between LUMINOUS and LUMINOUS3D

LUMINOUS3D has several other differences with LUMINOUS. For example, in Luminous the rays generated during ray tracing are stored in all subsequent structure files. In LUMINOUS3D this is not true. In order to save the rays in LUMINOUS3D the RAYTRACE parameter of the BEAM statement should be set to the name of a file where the results of the ray trace are to be stored

LUMINOUS3D also has an option to cause all metal regions to act as perfect reflectors. This option is enabled by specifying METAL.REFLECT on the BEAM statement.

Finally, LUMINOUS3D allows the user to simulate periodicity with respect to ray tracing by specifying PERIODIC on the BEAM statement. This causes rays exiting the sides of the device (perpendicular to the x-z plane) to re-enter the opposite side of the device.

Numerical Methods for 3D

DC Solutions

There are several differences between the default numerical methods applied in 2D ATLAS and thos applied in 3D ATLAS. Firstly, with respect to non-linear iteration strategies, the current version of the 3D simulatior does not support the BLOCK method. The NEWTON and GUMMEL iteration strategies are supported for 3D simulations whereas NEWTON, GUMMEL and BLOCK are all supported for 2D simulations.

In solving the linear subproblem, the default approach in 3D is to use an iterative solver. This is believed to be the most computationally efficient method for general 3D problems. In 2D the direct solver is used by default. The user may find it desirable to use direct methods in 3D problems due to improved convergence of computational efficiency. He may select the direct method by specifying DIRECT on the METHOD statement.

Also, in 3D there are two linear iterative solution methods available. The default is ILUCGS (incomplete lower-upper decomposition conjugate gradient system) and BICGST (bi-conjugate gradient stabilized). Historically, tests have shown that the current implementation of ILUCGS is slightly more stable that BICGST and it is the default iterative solver in 3D. The BICGST solver can be specified by the user by specifying BICGST on the METHOD statement.

Transient Solutions

In transient mode a semi-implicit scheme is used in addition to the default TR-BDF algorithm. This algorithm is recommended for complex simulations such as Single Event Upset. To select this method use:

```
METHOD HALFIMPL
```

Obtaining Solutions In 3D

ATLAS3D programs can perform DC and transient analysis in an equivalent manner to 2D. The SOLVE statement is used to define the solution procedure. The syntax used is exactly as that documented in the 'Getting Started' chapter.

Interpreting the Results From 3D

The Log files produced by 3D ATLAS can be plotted in TONYPLOT exactly as those that result from S-PISCES or BLAZE. The only difference is the units of the currents produced. Log files from 3D simulations save current in Amperes, whereas the 2D simulations use Amperes/micron.

The solution files produced by 3D ATLAS should be plotted using TONYPLOT3D. These files cannot be read directly into the 2D TONYPLOT program. TONYPLOT3D contains features that allow slices of the 3D structure to be made, which can be plotted in 2D TONYPLOT.

More Information

Many examples using 3D ATLAS have been installed on your distribution tape or CD. More information about the use of 3D ATLAS can be found by reading the text associated with each example.

Chapter 12: INTERCONNECT3D

Overview

INTERCONNECT3D solves Laplace's equation in a non-planar 3D region. It solves for the capacitance between metal conductors or the conductance of metal conductors.

Before reading this section users should read the 'Getting Started' section.

Model Description

Capacitance and Conductance Calculation

Capacitance and conductance analysis in INTERCONNECT3D is based on the solution of the Laplace equation to determine three dimensional potential and electric field distributions.

Laplace's equation is:

$$\nabla(\eta\nabla\psi) = 0 \qquad 12\text{-}1$$

where:

Ψ is electrostatic potential.

η is a parameter that may vary spatially and may be discontinuous. This parameter corresponds to dielectric permittivity (ε) in capacitance calculations, and to electrical conductivity (σ) in conductance calculations.

Boundary Conditions

Boundary conditions are needed to complete the mathematical specification of the problem. INTERCONNECT3D uses Dirichlet boundary conditions for contacts and Neumann boundary conditions for all other boundary planes.

When capacitances are being investigated, Equation 12-1 is solved with:

- Defined potentials on conductors
- η set to the dielectric permittivity (ε) of insulator regions

A normal electric field is set to 0 (except for contacts) at the external boundaries of the structure. This boundary condition is artificial. It assumes that each face of the structure has reflectional symmetry in every direction. This situation is illustrated in Figures 12-1 and 12-2. The structure simulated is shown in Figure 12-1 and the boundary conditions are shown in Figure 12-2.

Once the electric field distribution is known, the energy and/or total electric charge of the interconnection environment can be calculated. The capacitance matrix of the system is determined by calculating the energy and/or total electric charge for a sufficient number of different bias conditions. INTERCONNECT3D uses the energy conservation principle to calculate capacitance data because the energy conservation principle is more precise than the charge conservation principle.

A system of n conductors is described by a set of n(n-1)/2 capacitances. Every potential calculation with respect to a specific bias set yields one energy balance equation. Therefore, n(n-1)/2 3-D potential solutions are needed to obtain the minimal set of equations necessary to find all the capacitances of the system.

This computational effort is reduced drastically by:

1. Calculating and storing the potential distributions with respect to a set of n orthogonal bias sets.

2. Constructing further potential distributions through linear superposition of potential values from the existing solutions.

In this way, the number of full calculations required to solve the 3D problem is reduced from n(n-1)/2 calculations to n calculations.

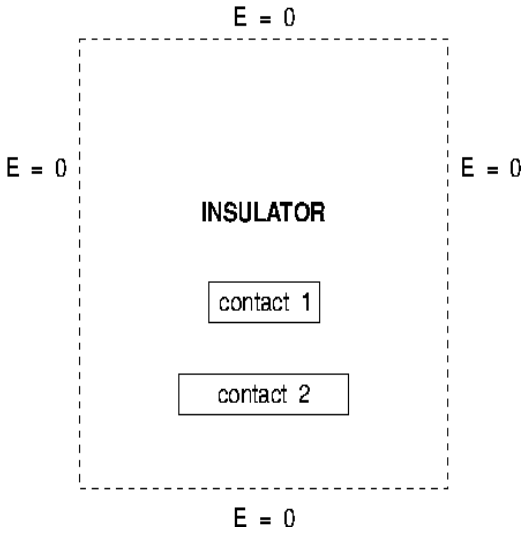


Figure 12-1: The structure to be simulated

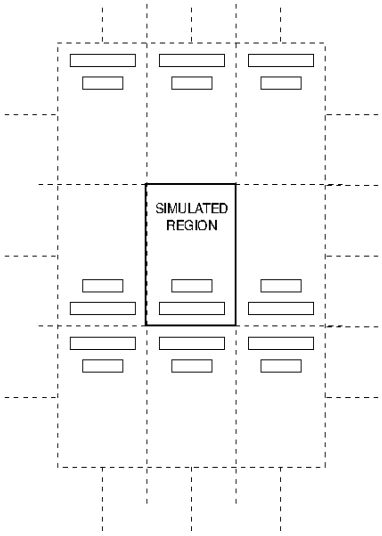


Figure 12-2: What INTERCONNECT3D actually simulates

These image conductors influence the simulation results if they are too close to the conductors in the simulated region. To minimize this influence, you must make the structure large enough so that there is enough distance between the image conductors to make the effect negligible.

When conductances are being investigated, equation 12-1 is solved in conducting regions with defined potentials on contacts and with η set to the electrical conductivity (σ). The local current density is calculated as $J = \sigma\nabla\psi$.

Structure Definition

Structures for INTERCONNECT3D can be defined in the same manner as described for DEVICE3D using the command language of ATLAS or an interface from DEVEDIT3D. The only exception is in electrode definition which is described in this section.

INTERCONNECT3D also has an interface to IC layout using MASKVIEWS.

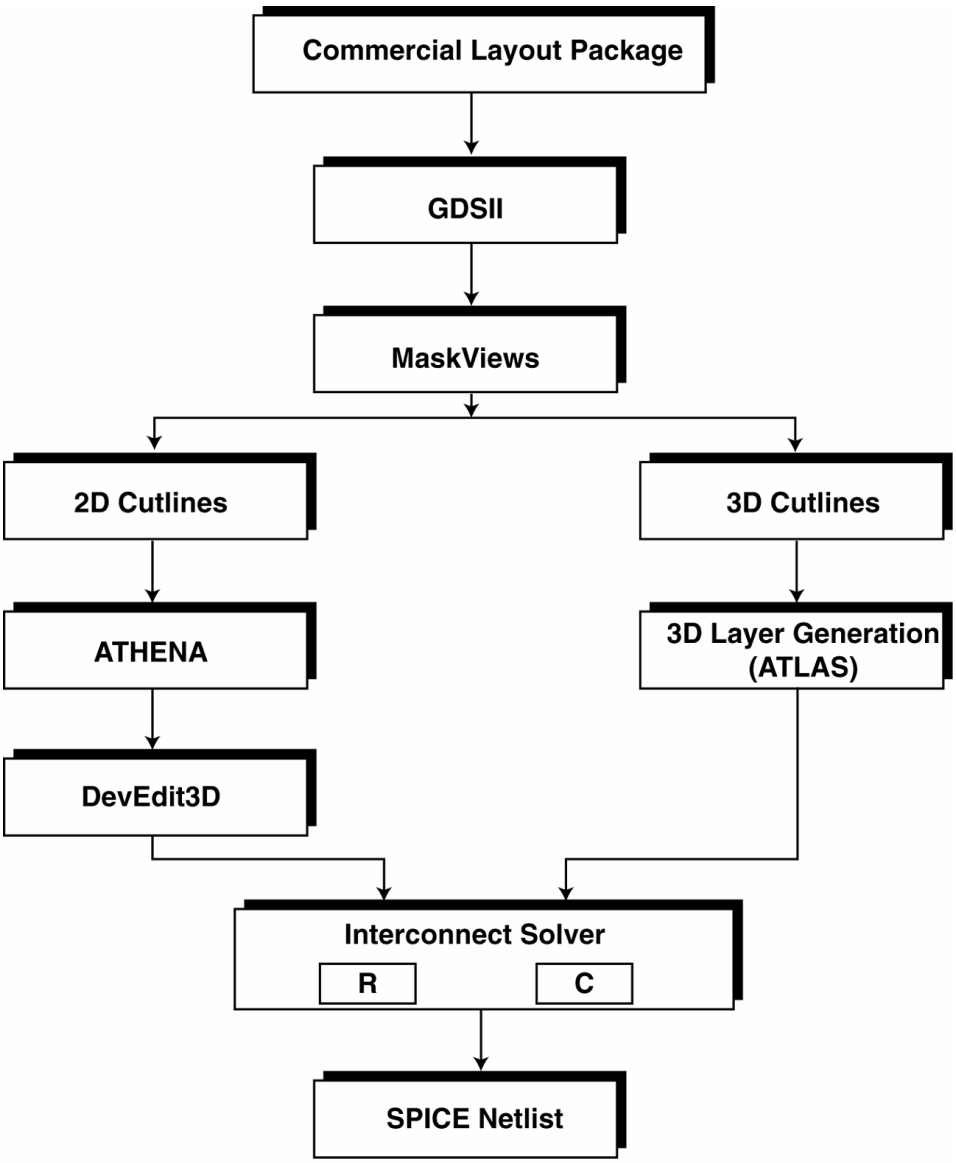


Figure 12-3: Information flow for parasitic extraction from IC layout

Figure 12-3 shows how the information from IC layout can be used in INTERCONNECT3D. The left hand option uses 2D process simulation and is described in the ATHENA and MASKVIEWS manuals. In this option the interface from DEVEDIT3D to INTERCONNECT3D is as documented in the "Getting Started" chapter.

The right hand branch of Figure 12-3 shows the direct layout to INTERCONNECT3D interface. This is documented below.

Tutorial For The MaskViews Interface

The MASKVIEWS-INTERCONNECT3D interface is a two step process. Firstly the mask data must be extracted and saved in MASKVIEWS. After this the layer structure of the interconnect is defined in ATLAS.

Saving Mask Data For INTERCONNECT3D

The operation of MASKVIEWS including the GDS-II interface is described in the MASKVIEWS documentation. This tutorial assumes the layout has been correctly entered in to MASKVIEWS.

1. Set the simulator in MASKVIEWS to ATLAS. Use the `Properties` menu to change the simulator to ATLAS. Press `Save` if you want to have ATLAS as the default interface from MASKVIEWS.
2. Press the button `Write_File` on the main MASKVIEWS window.
3. Go to the top left of the area of your layout you wish to simulate. Press the left mouse key and hold it down. Drag the mouse over the area you wish to load into INTERCONNECT3D.
4. A new menu will appear with the coordinates of the box you have drawn.
5. Enter the name of a cross-section file (e.g.. myfile.sec)
6. Press `Preview` on this menu.
7. A graphics window will appear with the mask layout in 3-D. Use this to check that your chosen area is correct. The resolution of complex mask scan be controlled using the resolution setting under the MASKVIEWS menu: `Define/Screen`.
8. Press `Write`. A file is written to your disk. This file will be used in INTERCONNECT3D.

Loading Mask Data Into INTERCONNECT3D

The section of the layout file saved by MASKVIEWS should be loaded into ATLAS through the `MESH` statement in the form

```
MESH NX=<INTEGER> NZ=<INTEGER> THREE.D LAYFILE=<FILENAME>
```

Note that the `THREE.D` parameter is required. The `LAYFILE` parameter specifies the name of the file saved by MASKVIEWS. Most importantly, notice that the `NY` parameter is not specified on the `MESH` statement. This is because the y-direction mesh is built up using the `ADDLAYER` statement.

The `ADDLAYER` statement makes a layer of the specified thickness in microns with the number of mesh points in the y-direction specified by the `DIVISION` parameter.

```
ADDLAYER thick=<VALUE> mask=<layer_name> division=<INTEGER VALUE>
```

The `MASK` parameter tells ATLAS which mask to use from the MaskViews layout. The layer name must correspond exactly with the layer name in MASKVIEWS.ATLAS assumes the drawn features are metallic regions and the clear regions are insulator. If no `MASK` parameter is used, ATLAS assumes the layer is completely formed from insulator.

Defining Electrodes

Electrode specification in INTERCONNECT3D is different from other ATLAS simulations. Typical structures for INTERCONNECT3D consist of many metal conductors embedded in blocks of insulator. Some of the metal conductors are connected to make an interconnect. Each interconnect needs to be

defined by a unique electrode number or name. In INTERCONNECT3D, connectivity between conductors is done automatically with no user intervention required.

There are several methods electrodes can be defined for INTERCONNECT3D:

- Define electrodes as usual in ATHENA, DEVEDIT3D or ATLAS. See 'Getting Started' for details. This method is typical if the DEVEDIT3D interface is used for conductance calculations.
- Do not define any electrodes. In this case the program finds connected blocks of metal for each interconnect. Each interconnect is assigned a different electrode number. The position of a point in each conductor is printed to the run-time output to enable users to tell the number of each interconnect. This method is typical when the MASKVIEWS interface is used.
- Specify the point which is inside any conductor using the ELECTRODE statement.

```
ELECTRODE NUM=<integer> X.PNT=<val> Y.PNT=<<val> Z.PNT=<val>\
```

All the metal conductors of the interconnect structure connected to that point will assume the same electrode number. X.PNT and Z.PNT have no defaults, Y.PNT has a default of zero.

Specify parallelopiped, which intersects electrode, with boundaries (X.MIN,X.MAX)(Y.MIN,Y.MAX), (Z.MIN,Z.MAX), in the ELECTRODE statement; then all the connected points of interconnect structure will refer to the same electrode.

```
ELECTRODE NUM=<integer> X.MIN=<val> Y.MIN=<val> Z.MIN=<val> \
          X.MAX=<val> Y.MAX=<val> Z.MAX=<val>
```

In this case default values of these parameters differ from the usual ATLAS case. They are:

```
X.MIN - has no default;
X.MAX - has no default;
Y.MIN = 0;
Y.MAX = <layer_thickness>;
Z.MIN = 0;
Z.MAX = 0;
```

Model And Material Parameter Selection

Conductivity Calculations

In conductivity calculations, the conductance of each region needs to be set using the GCOND parameter of the MATERIAL statement. For example:

```
MATERIAL REGION=2 GCOND=1.E3
```

To select that a conductance calculation is required the following statement is used:

```
MODEL EX.RES
```

Capacitance Calculations

For capacitance calculation, the relative permittivity of insulator regions can be set using the MATERIAL statement. If the DEVEDIT3D interface is used this can be done using:

```
MATERIAL REGION=<integer> PERM=<value>
```

If the MASKVIEWS interface is used, INTERCONNECT3D assumes that all insulators are silicon dioxide and have a relative permittivity of 3.9. To override this, the MATERIAL statement is used. For example to set a nitride dielectric between two conductors in layers M1 and M2:

```
ADDLAYER MASK=M1      THICK=1.0  DIV=5
ADDLAYER              THICK=1.0  DIV=5
ADDLAYER MASK=M2      THICK=1.0  DIV=5
MATERIAL LAYER=2      PERM=7.2
```

The value of the LAYER parameter is counted from the bottom upwards.

To select that a capacitance calculation is required the following statement should be included in the command file.

```
MODELS EX.CAP
```

Applying CD Variations and Misalignment

The ADDLAYER statement supports the use of parameters to model variations in the interconnect structure. The DELTA.CD.X and DELTA.CD.Z parameters are used to define a offset to the size of any metal feature on that layer in the indicated direction. The units are microns. Negative values will increase the metal width with the X or Z direction, positive values will decrease it.

Similarly the parameters MISALIGN.X and MISALIGN.Z are used to misalign layers relative to the positions in the Maskviews section file. Separate misalignments in X and Z can be specified to model the worst case alignment of multi-layer interconnects. The default values are 0.0.

Numerical Methods

No special numerical methods are required for either conductance or capacitance simulations. No METHOD statement is required.

Obtaining Solutions

A solution is obtained for both capacitance and conductance modes with the simple statement:

```
SOLVE OUTF=<filename>
```

Only one SOLVE statement is allowed in each run of INTERCONNECT3D.

Interpreting the Results

The output of INTERCONNECT3D simulation consist of a SPICE sub-circuit netlist of the capacitance or conductance of the structure. For capacitance calculations this is the capacitance between each pair of conductors in Farads. For conductance calculations it is the resistance between each pair of electrodes in ohms.

The three dimensional structure and potential distribution can be saved in an output structure file by setting the OUTFILE parameter in statement. This structure can be plotted using TONYPLOT3D.

More Information

Examples using INTERCONNECT3D have been installed on your distribution tape or CD. More information about the use of INTERCONNECT3D can be found by reading the text associated with each example.

[This page intentionally left blank]

Chapter 13: THERMAL3D

Overview

THERMAL3D solves the steady-state heat equation to find the equilibrium temperature distribution in planar and nonplanar three-dimensional structures. The user specifies the heat sinks and sources, and also chooses from several temperature-dependent models for thermal conductivity within each region. A typical application for THERMAL3D is a package simulation for a power circuit or III-V ICs.

Before reading the following sections, users are advised to read the 3D device simulation chapter.

3D Structure Generation

THERMAL3D supports structure defined on 3-D prismatic meshes. Structures may have arbitrary geometries in two dimensions and consist of multiple slices in the third dimension. There are two methods for creating a 3-D structure that can be used with THERMAL3D. One is through the command syntax of ATLAS, and the other through an interface to b. Please see documentation on both of these methods in the DEVICE3D chapter.

Defining Heat Sources

Heat sources are identified with regions in the 3D structure. Regions are defined in the manner documented in the "Getting Started" and 3D device simulation chapters. A region has a unique number, which is used to identify the region on the MATERIAL statement. The POWER parameter of the MATERIAL statement is used to set the power of the heat source in watts.

```
MATERIAL REGION=2 POWER=0.35
```

The POWER parameter must be set on the SOLVE statement if the user wishes to step the power through a range of values. See the section "OBTAINING SOLUTIONS IN THERMAL3D" in this chapter for the proper syntax.

Defining Heat Sinks

Heat sinks are identified as electrodes in the 3-D structure. Heat sink areas should be defined as electrodes in the manner documented in the "Getting Started" and "3-D Device Simulation" chapters. Each electrode (heat sink) has a unique number, which is used to set the temperature on the heat sink during simulation. Setting the temperature of the heat sink is documented in the 'Obtaining Solutions' section of this chapter.

Model and Material Parameter Selection

Thermal Simulation Model

To obtain steady-state solutions for the temperature distribution, THERMAL3D solves Poisson's equation for temperature:

$$\nabla(k(T)\nabla T) = q \qquad 13\text{-}1$$

where T represents the steady state temperature, k the temperature-dependent thermal conductivity, and q the power generation per unit volume in the medium (heat sources).

The prescribed temperatures at the heat sinks form boundary conditions for equation 13-1.

The solution of the heat equation by THERMAL3D is invoked by the following syntax:

```
MODELS THERMAL
```

Setting Thermal Conductivity

The value of thermal conductivity k, for each region should be specified in the MATERIAL statement. Because thermal conductivity is, in general, temperature dependent; the following four models are available:

$$k(T) = k0 \qquad (W/(cm)/K) \qquad 13\text{-}2$$

$$k(T) = (k0)/(T/300)^{n} \qquad (W/(cm)/K) \qquad 13\text{-}3$$

$$k(T) = 1/(A + B*T + C*T^{2}) \qquad (W/(cm)/K) \qquad 13\text{-}4$$

$$k(T) = E/(T - D) \qquad (W/cm/K) \qquad 13\text{-}5$$

Table 13-1. User-Specifiable Parameters For Equations 13-2 to 13-5

Symbol	Parameter	Statement
k0	TC.CONST	MATERIAL
n	TC.NPOW	MATERIAL
A	TC.A	MATERIAL
B	TC.B	MATERIAL
C	TC.C	MATERIAL
D	TC.D	MATERIAL
E	TC.E	MATERIAL

To choose the model in Equation 13-2, the parameter, TCON.CONST should be specified (default). To choose the model in Equation 13-3, the paramete,r TCON.POLYN should be specified. To choose the model in Equation 13-5, the parameter, TCON.RECIP should be specified.

Example

```
MATERIAL REGION=2 TCON.POWER TC.CO=1.0 TC.NPOW=1.2
```

means that the temperature-dependent thermal conductivity for region #2 is expressed as

$$K(T) = \frac{1}{(T/300)^{1.2}} \qquad 13\text{-}6$$

Suggested Parameters For Thermal Conductivity

There are no pre-set material defaults in THERMAL3D. The following values are recommended for thermal conductivity of GaAs and Si.

For GaAs, use model B with

```
MATERIAL TCON.POWER TC.CO=0.44 TC.POW=1.25
```

For Si, use model B, C, or D with

```
(model B)MATERIAL TCON.POWER TC.CO=1.55 TC.POW=-1.33
(model C)MATERIAL TCON.POLYN TC.A=0.03 TC.B=1.56e-3 TC.C=1.65e-6
(model D)MATERIAL TCON.RECIP TC.E=320 TC.D=80
```

Numerical Methods

No special numerical methods are required for thermal simulation. A METHOD statement with no parameters is assumed by default.

Obtaining Solutions In THERMAL3D

The SOLVE statement in used in THERMAL3D for heat-flow solutions much the same as it is used in other ATLAS simulations involving electrical biases. The temperature in kelvin on each heat sink is used to prescribe the boundary condition temperatures. For example,

```
SOLVE T1=300 T2=500
```

sets the temperature at 300 K and 500 K on heat sink #1 and #2 respectively.

Multiple SOLVE statements are allowed in THERMAL3D. This is useful for obtaining solutions for several combinations of heat sinks and thermal power sources. A range of solutions can also be obtained by stepping the value of a heat sink or power source. For example,

```
SOLVE T1=300 POWER2=0.35 POWER3=0.4 NSTEPS=5 STEPREGION=3 \
POWERFINAL=0.8 OUTFILE=thermal_out0
```

increments the thermal power source in region #3 from 0.4 watts to 0.8 watts in 5 steps and:

```
SOLVE T1=300 POWER2=0.35 NSTEPS=3 ELECTRODE=1 \
TEMPFINAL=600 OUTFILE=thermal_out0
```

increments the temperature on electrode #1 from 300 K to 600 K in 3 steps. Thermal power and temperature can be simultaneously sweep.

If more than one region power is specified during a sweep, the region to be stepped must be specified by STEPREGION=# as shown in the first example above. If the STEPREGION parameter is not specified, the smallest numerical value of POWER# is stepped.

During temperature and power sweeps, the output filename is modified according to the following rule: The rightmost character is incremented using the sequence 0-9, A-Z, a-z. When a character is incremented beyond ‘z’ the character is set to ‘0’> and the character to the left (smaller than ‘z’) is incremented.

Complete syntax information for the SOLVE statement can be found in Chapter 15, "Statements".

The SOLVE statement is used in THERMAL3D for heat flow solutions much as it is used in other ATLAS simulations for electrical biases. The temperature in Kelvin on each heat sink in the device *must* be set on the SOLVE statement. The parameter, Tn, where n is the number of the heat sink, is used to set the temperature. For example:

```
SOLVE T1=300 T2=500
```

This sets 300K on heat sink #1 and 500K on heat sink #2. Only one SOLVE statement is allowed in any THERMAL3D input file.

Interpreting The Results From THERMAL3D

The output of thermal simulation consists of the minumum and maximumcalculated temperature of each region and its location. The three-dimensional temperature distribution can be saved in an ouput structure file by setting the OUTFILE parameter in the SOLVE statement and visualized using TONYPLOT3D.

The output of thermal simulation consist of the minimum and maximum calculated temperature of each region and its location. The three dimensional temperature distribution can be saved in a output structure file by setting the OUTFILE parameter in the SOLVE statement and visualized using TONYPLOT3D.

More Information

Many examples using THERMAL3D have been installed on your distribution tape or CDROM. More information about the use of THERMAL3D can be found by reading the text associated with each example.

Chapter 14: Numerical Techniques

Overview

This chapter describes the overall process of obtaining a numerical solution, the subtasks involved, and the options and defaults available in ATLAS.

Users do not need to master this material in order to use ATLAS. Chapter Two presents the information about numerical techniques that is needed by most users. This chapter provides additional information that will mainly be of interest to advanced users.

Numerical Solution Procedures

Semiconductor device operation is modeled in ATLAS by a set of anywhere from one to six coupled, non-linear, partial differential equations (PDEs). ATLAS produces numerical solutions of these equations by calculating the values of unknowns on a mesh of points within the device. An internal discretization procedure converts the original, continuous model to a discrete non-linear algebraic system that has approximately the same behavior. The set of PDEs, the mesh and the discretization procedure determine the non-linear algebraic problem that must be solved.

The non-linear algebraic system is solved using an iterative procedure that refines successive estimates of the solution. Iteration continues until the corrections are small enough to satisfy convergence criteria, or until it is clear that the procedure is not going to converge. The non-linear iteration procedure starts from an initial guess. The corrections are calculated by solving linearized versions of the problem. The linear subproblems are solved by using direct techniques or iteratively.

Different solution procedures exhibit different behavior with respect to convergence, accuracy, efficiency, and robustness. The two main aspects of convergence are whether a solution is obtained, and if so, how rapidly it is approached. Accuracy is how closely the computed solution approximates the true solution. Efficiency is the time required to produce a solution. Robustness is the ability to converge for a wide range of structures, using meshes and initial guess strategies that are not optimum.

When solving general systems of non-linear equations there are no guarantees that any particular method will always work. It is also the case that different methods can work better for different problems. Fortunately there is now a lot of practical experience concerning the numerical techniques that are effective for device simulation. This practical experience has been captured in ATLAS in the form of default methods and parameters that work well in almost all circumstances. This chapter provides advanced information of interest to users who want to change the defaults.

Meshes

The specification of meshes involves a trade-off between the requirements of accuracy and numerical efficiency. Accuracy requires a fine mesh that can resolve all significant features of the solution. Numerical efficiency requires a coarse mesh that minimizes the total number of grid points. This trade-off between accuracy and numerical efficiency is frequently a source of problems for beginners. Fortunately, enough experience to define reasonable meshes is soon acquired.

ATLAS uses triangular meshes. Some triangulations yield much better results than others. Mesh generation is still an inexact science, but guidelines and heuristics for defining satisfactory meshes exist. ‘Good’ triangulations have the following features:

- They contain enough points to provide the required accuracy.
- They do not contain too many unnecessary points that impair efficiency.

- They avoid, or at least minimize, the number of obtuse triangles. Obtuse triangles tend to impair accuracy, convergence, and robustness.
- They avoid, or at least minimize, the number of long, thin triangles. These triangles also tend to impair accuracy, convergence, and robustness.
- They allow the average size of triangles to change smoothly in transition from a region where very small triangles must be used to a region where the use of much larger triangles is acceptable.

The error associated with a mesh can be investigated systematically by repeating a calculation using a sequence of finer meshes. This is very time consuming and is hardly ever done. The typical approach is to adequately resolve structural features, including doping, with an initial or base mesh, and then add nodes as required to resolve significant features of the solution. The insertion of additional nodes (regridding) is normally done by the program, using criteria specified by the user.

The initial mesh used by ATLAS can be specified in several ways: it can be "inherited" from ATHENA, it can be constructed using DEVEDIT, or it can be specified using the ATLAS command language. Meshes can be refined using ATLAS commands, or using DEVEDIT. The remainder of this section will focus on the capabilities available using ATLAS commands. The capabilities provided by DEVEDIT are documented in the VWF INTERACTIVE TOOLS manual.

There are limits on the maximum number of nodes that can be specified. Two-dimensional ATLAS simulations may have up to 9,600 nodes. Three-dimensional simulations may have up to 200,000 nodes, 400,000 elements, with no more than 20,000 in a single plane and a maximum of 200 planes in the z direcction. Most devices can be adequately simulated in two dimensions using meshes that contain from several hundred to around 3000 nodes.

Mesh Regridding

The REGRID statement supports refinement of regions of the mesh according to specified criteria. Refinement can occur when a specified solution variable exceeds some value, or when the change in that variable across a triangle exceeds a value. The variable can be any of the key quantities in a problem, such as potential, carrier concentration, doping concentration, or electric field.

The regrid algorithm searches the initial grid for triangles that meet the criterion specified for refinement. Each triangle that is identified is divided into four congruent subtriangles. Grid quantities (doping, potential, carrier concentrations, and so forth) are interpolated onto the new nodes, using linear or logarithmic interpolation, as appropriate for that quantity. The initial grid is referred to as being "on level 0" and the new triangles are referred to as "on level 1". After all level 0 triangles have been examined, the same procedure is applied to level 1 triangles, and any subtriangles of level 1 become "level 2"> triangles. The grid is checked for consistency at each level and is updated to avoid abrupt changes of size from one triangle to the next. The regrid process continues until no more triangles meet the refinement criteria, or until a specified maximum level of refinement is reached. Grids used in practice are often coarser than is required to meet desirable refinement criteria, and so the maximum level is the key factor in determining the size of the grid after refinement.

The MAX.LEVEL parameter of the REGRID statement is used to limit the amount of refinement at each step. By default, ATLAS sets the maximum level equal to one more than the highest level in the existing mesh. To update a coarse region without regridding the finer regions, after a mesh has already been refined several times, set the maximum level below the level of the finer regions in the existing grid.

If several levels of regrid are performed in immediate succession, interpolated data is used to make the refinement decisions at higher levels. Since semiconductor problems are non-linear, this interpolation may not produce satisfactory results. It is often a good idea to calculate a new solution between regrid operations, i.e. to regrid only one level at a time and obtain a new solution each regrid operation.

Two popular choices of quantities to be used for mesh refinement are potential and doping. Ideally, variations of electrostatic and quasi-Fermi potentials across an element would be limited to no more

than kT/q, and variations in doping would be no more than a factor of 5 or so. In practice the refinement criteria are often significantly coarser: around 10-20 kT/q for potential, and two to three orders of magnitude for doping. In high level injection situations, it is a good idea to regrid when the value of minority carrier concentration exceeds the local doping concentration.

Mesh Smoothing

Although every step of grid generation can introduce obtuse triangles, two steps in particular can cause problems. The first is that distorting a rectangular mesh introduces a very large number of obtuse elements. The second is that when regridding a rectangular grid that contains triangles with an aspect ratio of 4:1 or greater, very obtuse triangles are created in the transition region between high and low grid density. The REGRID statement allows several procedures to be used when dealing with poorly shaped elements such as obtuse triangles.

The two techniques are node smoothing and triangle smoothing. With node smoothing, several iterative passes are carried out during which each node is moved to a position which improves the angles of the triangles surrounding it. Node smoothing should only be used for grids that are already irregular. If node smoothing is used for nearly rectangular grids, it may significantly degrade the quality of the mesh.

With triangle smoothing (which is also referred to as diagonal flipping), each adjoining pair of triangles is examined. If appropriate, the diagonal of the quadrilateral is flipped to stabilize the discretization. The diagonal is never flipped when two elements are composed of different materials. When elements are of the same material but have different region numbers, the user can specify whether or not to flip the diagonals.

Triangle smoothing is desirable in almost all cases, and should be performed on both the initial grid and on subsequent regrids. The only exception to this rule arises from a undesirable interaction of three elements: regrid, high aspect ratio triangles, and smoothing. This situation frequently occurs in gate oxide regions that involve long, thin triangles. In these cases, smoothing may produce large triangles surrounded by many smaller triangles, giving the appearance of a hole in the mesh. To overcome this, the smoothing command SMOOTH=4 can be used to limit the formation of the large triangles.

Discretization

The Discretization Process

The discretization process yields relationships between the variables defined at mesh points. In order to be useful, a discretization must be consistent, i.e. it must limit to the underlying equation in the limit that the mesh size tends to zero. All of the discretizations used in ATLAS have this property. Different discretizations can have different properties with respect to accuracy. The most important measure of accuracy is the order of the scheme, i.e. how errors scaled as the difference between mesh points tends to zero. Discretization schemes used in device simulation are often second order, i.e. as the mesh becomes very fine the discretization error varies as the square of the separation between mesh points.

The discretizations implemented in ATLAS use the box integration method to approximate differential operators on a general triangular grid. Each equation is integrated over a small polygon which encloses each node. The set of all polygons completely covers the solution domain. The integration procedure equates fluxes into a polygon with sources and sinks inside the polygon, which means that quantities that are conserved physically are also conserved by the numerical solution.

The fluxes must be discretized carefully for the carrier continuity and energy balance equations, otherwise nonphysical oscillations and negative carrier concentrations and temperatures may arise. Scharfetter and Gummel[35] introduced approximations for current densities that overcome this

problem. Generalizations of this approach are used in ATLAS for the discretization of current densities and energy fluxes.

Non-Linear Iteration

The non-linear solution method, and associated parameters such as iteration and convergence criteria, are specified in the METHOD statement. Non-linear iteration solution methods are specified in the METHOD statement using the parameters NEWTON, GUMMEL, or BLOCK. Combinations of these parameters may also be specified. In order to understand the effect of these parameters, it is helpful to briefly review how numerical solutions are obtained.

The non-linear algebraic system that results from discretization on a mesh is solved iteratively starting from an initial guess. Linearized subproblems are set up and solved. These provide corrections that are used to update the current estimate of the solution. Different sequences of linear subproblems correspond to different non-linear iteration strategies. Iteration continues until convergence criteria are met, in which case the solution is accepted; or until a preset maximum allowable number of iterations is reached, in which case a different technique is tried or the solution procedure is abandoned. When a solution fails to converge, a user normally tries a different grid, a different initial guess strategy, or a different non-linear iteration technique.

Newton Iteration

Each iteration of the Newton method solves a linearized version of the entire non-linear algebraic system. The size of the problem is relatively large, and each iteration takes a relatively long time. However, the iteration will normally converge quickly (in about three to eight iterations) so long as the initial guess is sufficiently close to the final solution. Strategies that use automatic bias step reduction in the event of non-convergence loosen the requirement of a good initial guess. Newton's method is the default for drift-diffusion calculations in ATLAS. There are several calculations for which ATLAS *requires* that Newton's method is used. These are: DC calculations that involve lumped elements; transient calculations; curve tracing; and when frequency-domain small-signal analysis is performed.

The Newton-Richardson method is a variant of the Newton iteration that calculates a new version of the coefficient matrix only when slowing convergence demonstrates that this is necessary. An automated Newton-Richardson method is available in ATLAS, and improves performance significantly on most problems. The automated Newton-Richardson method is enabled by specifying the AUTO parameter of the METHOD statement.

If convergence is obtained only after many Newton iterations, the problem is almost certainly poorly defined. The grid may be very poor (i.e. it contains many obtuse or high aspect ratio triangles), or a depletion region may have extended into a region defined as an ohmic contact, or the initial guess may be very poor.

Gummel Iteration

Each iteration of Gummel's method solves a sequence of relatively small linear subproblems. The subproblems are obtained by linearizing one equation of the set with respect to its primary solution variable, while holding other variables at their most recently computed values. Solving this linear subsystem provides corrections for one solution variable. One step of Gummel iteration is completed when the procedure has been performed for each independent variable. Gummel iteration typically converges relatively slowly, but the method will often tolerate relatively poor initial guesses. The Gummel algorithm can not be used with lumped elements or current boundary conditions

Two variants of Gummel's method can improve its performance slightly. These both limit the size of the potential correction that is applied during each Gummel loop. The first method, called damping, truncates corrections that exceed a maximum allowable magnitude. It is used to overcome numerical ringing in the calculated potential when bias steps are large (greater than 1V for room temperature calculations). The maximum allowable magnitude of the potential correction must be carefully

specified: too small a value slows convergence, while too large a value can lead to overflow. The DVLIMIT parameter of the METHOD statement is used to specify the maximum allowable magnitude of the potential correction. By default the value of this parameter is 0.1 V. Thus, by default Gummel iterations are dampped. To specify undamped Gummel iterations, the user should specify DVLIMIT to be negative of zero.

The second method limits the number of linearized Poisson solutions per Gummel iteration, usually to one. This leads to under-relaxation of the potential update. This "single-Poisson" solution mode extends the usefulness of Gummel's method to higher currents. It can be useful for performing low current bipolar simulations, and simulating MOS transistors in the saturation region. It is invoked by specifying the SINGLEPOISSON parameter of the METHOD statement.

Block Iteration

ATLAS offers several block iteration schemes that are very useful when lattice heating or energy balance equations are included. Block iterations involves solving subgroups of equations in various sequences. The subgroups of equations used in ATLAS have been established as a result of numerical experiments that established which combinations are most effective in practice.

In nonisothermal drift-diffusion simulation, specifying the BLOCK method means that Newton's method is used to update potential and carrier concentrations, after which the heat flow equation is solved in a decoupled step.

When the carrier temperature equations are solved for a constant lattice temperature, the BLOCK iteration algorithm uses Newton's method to update potential and concentrations. The carrier temperature equation is solved simultaneously with the appropriate continuity equation to update the carrier temperature and again carrier concentration.

When both the heat flow equation and the carrier temperature equations are included, the BLOCK scheme proceeds as described previously for the carrier temperature case, and then performs one decoupled solution for lattice temperature as a third step of each iteration.

Combining The Iteration Methods

It is possible to start with the GUMMEL scheme and then switch to BLOCK or NEWTON if convergence is not achieved within a certain number of iterations. One circumstance where this can be very helpful is that Gummel iteration can refine initial guess to a point from which Newton iteration can converge. The number of initial GUMMEL iterations is limited by GUM.INIT.

It can also be desirable to use BLOCK iteration and then switch to NEWTON if convergence is not achieved. This is the recommended strategy for calculations that include lattice heating or energy balance. The number of initial BLOCK iterations is limited by NBLOCKIT

Any combination of the parameters GUMMEL, BLOCK and NEWTON may be specified on the METHOD statement. ATLAS will start with GUMMEL if it is specified. If convergence is not achieved within the specified number of iterations, it will then switch to BLOCK if BLOCK is specified; if convergence is still not achieved the program will then switch to NEWTON.

Solving Linear Subproblems

The linear subproblems generated by non-linear iteration can be solved by direct or iterative methods. Direct methods produce solutions in a predictable number of arithmetic operations. Solutions are affected by roundoff error, but are otherwise exact. Iterative methods obtain solutions by making a series of corrections to an initial guess. Iteration proceeds until the calculated corrections satisfy specified convergence criteria. The results are not exact to within roundoff error, and convergence is not guaranteed for general problems. However, iterative methods can be more efficient than direct methods for large linear systems, and generally require less memory.

There are mathematical proofs regarding the types of linear system for which iterative techniques converge. All common iterative schemes converge for linear systems that are symmetric positive definite (SPD). The linearized form of Poisson's equation is of this type. The continuity equations are not SPD, but they can be reliably solved by modern, advanced iterative methods.

The size and structure of the coefficient matrix of the linear system plays an important role in the choice of direct or iterative methods. The overall problem size is determined by the number of variables per node (m) and the number of nodes (n). The number of unknowns is m x n. The linear sub problems associated with Newton iteration have a coefficient matrix with $(m \times n)^2$ elements. Each linearized subproblem which is used in Gummel iteration has a coefficient matrix with n^2 elements.

For practical 2-D device simulation problems, the number of elements in the coefficient matrix is typically between 10^5 and 10^8. Fortunately the matrices are sparse, i.e. most of the entries are zero, and need not be stored explicitly. The sparsity arises because the variables at each node are coupled to only a few neighboring nodes. Special direct techniques are available for solving sparse matrices.

Direct techniques are preferred for relatively small problems. They provide reliable solutions that are exact to within roundoff error in a time that is predictable. Iterative techniques are preferred for very large problems because they are faster and require less memory. Direct techniques for solving sparse matrices have a competitive performance for the problem sizes typically encountered in 2-D device simulation, and are used by ATLAS to solve most of the linear subproblems that arise.

Convergence Criteria for Non-linear Iterations

After a few non-linear iterations, the errors will generally decrease at a characteristic rate as the iteration proceeds. Non-linear iteration techniques typically converge at a rate that is either linear or quadratic. The error decreases linearly when Gummel iteration is used, i.e. it is reduced by about the same factor at each iteration. For Newton iteration the convergence is quadratic, i.e., small errors less than one are approximately squared at each iteration. The non-linear iteration is terminated when the errors are acceptably small. The conditions required for termination are called convergence criteria. Much effort has gone into developing reliable default convergence criteria for ATLAS. The default parameters work well for nearly all situations, and most users will never need to change them.

Error Measures

A single positive number that characterizes error is obtained by taking a norm of the errors associated with each unknown. The quantity that ATLAS tries to reduce to zero is the difference between the left and right hand sides of the equation. It is natural to use this quantity as the measure of the error. The associated error norm is called the right hand side (RHS) norm. The units of the RHS norm are C/μm for the Poisson equation, and A/μm for the continuity equations.

Carrier Concentrations and CLIM.DD (CLIMIT)

Another measure of error is provided by the size of the calculated corrections for each unknown. Since the updates are the unknown "xs" at each step, this is called the X norm. Potential updates are measured in units of kT/q. Updates to carrier concentrations are measured relative to the previous value at the point. This relative error (ε_C) is defined as:

$$\varepsilon_C = \max_m \frac{C_m^{K+1} - C_m^K}{\max(C_0, C_m^K)} \qquad 14\text{-}1$$

where:

C = n or p, for electrons and holes respectively.

m is the node identifier.

C_0 is a characteristic concentration.

K is the iteration number

C_0 is specified as or as CLIM.DD or as CLIM·c*,

$$\text{CLIM.DD} = \text{CLIMT}\cdot c^* . \qquad 14\text{-}2$$

where:

$$c^* = \sqrt[4]{N_c N_v} \qquad 14\text{-}3$$

CLIM.DD (or CLIMIT) is specified in the METHOD statement.

It is difficult to specify a reasonable default value for the parameter CLIM.DD in all situations. In many difficult cases, the round-off numerical errors do not allow for the resolution of very low concentrations. The default value of CLIMIT is set at 10^4 (the corresponding default value for CLIM.DD in Silicon i s $4.5\cdot10^{13}cm^{-3}$). In simulation of breakdown, a lower value of CLIM.DD (~10^8cm^{-3} for Silicon diodes) should be specified; otherwise, a "false" solution may be obtained.

Discussion of CLIM.EB

To estimate errors in the lattice temperature equation and the energy balance equations, corresponding RHS norms and X norms are calculated in ATLAS. Updates to temperature are measured relative to some characteristic value of temperature. The parameter CLIM.EB, can be viewed as a regularization parameter for the case of very small electron or hole densities in the energy balance equations. The CLIM.EB parameter specifies the minimum value of concentration for which the relaxation term in the energy balance equation will be properly resolved. The temperatures for points where the concentration is much less than CLIM.EB are equal to the lattice temperature. The units of CLIM.EB are cm^{-3} and the default is 0.0.

Terminal Current Criteria

Another qualification for convergence is derived from the relative changes in terminal currents and the satisfaction of total current conservation. This qualification can be expressed as the simultaneous satisfaction of the following conditions:

$$\left|I_i^{K+1} - I_i^K\right| \leq E_1\left|I_i^{K+1}\right| + E_2 \qquad 14\text{-}4$$

and

$$\left|\sum I_i^{K+1}\right| < 0.01 \max\left(\left|I_i^{K+1}\right|, E_2\right)$$

$$I = 1, nc \qquad 14\text{-}5$$

where

I_i is the current through contact i

K is the iteration number

E_1 and E_2 are specified tolerances

nc is the number of contacts

Summary of Termination Criteria

A summary of the termination criteria that is enough for most purposes will now be given. Detailed reference information is provided in the next section.

The non-linear iteration is terminated when one of the following four criteria is satisfied.

1. The X norm for every equation falls below a specified tolerance. The specified tolerances for X norms are:

   P_{tol}^{x} - for potential equation;

   C_{tol}^{x} - for concentration equations;

   TL_{tol}^{x} - for lattice temperature equation;

   TC_{tol}^{x} - for carrier temperature equation.

2. The RHS for all equations and X norms for energy balance falls below a specified tolerance. The specific tolerances are:

   P_{tol}^{r} - for potential equations;

   C_{tol}^{r} - for concentration equations;

   TL_{tol}^{r} - for lattice temperature equations;

   TC_{tol}^{r} -for carrier temperature equations.

3. For every equation either the X norm or the RHS norm falls below a specified tolerance. In this case, both the XNORM and RHSNORM parameters must be specified true.
4. If either 1. or 2. or 3. criterion is fulfilled for weaker values of tolerances, i.e. for specified tolerances multiplied by the parameter W, and current criteria 14-4 and 14-5 are satisfied.

To exclude the X-norm criterion, ^XNORM should be specified in the METHOD statement; to exclude RHS-norm criterion, ^RHSNORM should be specified; to exclude the current criterion E1 and E2 should be made very small. All the above mentioned tolerances can be changed simultaneously by specifying the relaxation factor TOL.RELAX in the METHOD statement.

Table 14.1. User-Specifiable Parameters for Convergence Criteria			
Symbol	**Statemen**	**Parameter**	**Default**
P_{tol}^{x}	METHOD	PX.TOL	10^{-5}
C_{tol}^{x}	METHOD	CX.TOL	10^{-5}
TL_{tol}^{x}	METHOD	TLX.TOL	10^{-5}
TC_{tol}^{x}	METHOD	TCX.TOL	10^{-5}
P_{tol}^{r}	METHOD	PR.TOL	$5.0x10^{-26}$
C_{tol}^{r}	METHOD	CR.TOL	$5.0x10^{-18}$
TL_{tol}^{r}	METHOD	TLG.TOL	100
TC_{tol}^{r}	METHOD	TCR.TOL	100
E_1	METHOD	IX.TOL	$2.0x10^{-5}$
E_2	METHOD	IR.TOL	$5.0x10^{-15}$
W	METHOD	WEAK	200
TOL.RELAX	METHOD	TOL.RELAX	1
XNORM	METHOD	XNORM	TRUE
RHSNORM	METHOD	RHSNORM	TRUE
CLIMIT	METHOD	CLIMIT	10^{4}
CLIM.DD	METHOD	C_0	$4.5x10^{13}$
CLIM.EB	METHOD	CLIM.EB	0

Detailed Convergence Criteria

Only in very difficult situations is more detailed information concerning error estimation and the specification of convergence criteria needed. The material is organized by algorithm.

Convergence Criteria For Gummel's Algorithms

Relative update errors are defined as follows:

for potential

$$\varepsilon_v = \frac{\max_m \left|\varphi_m^{K+1} - \varphi_m^K\right|}{max(1, \varphi_{mmax}^{K+1})}, \qquad 14\text{-}6$$

where mmax is the node where $\left|\varphi_m^{K+1} - \varphi_m^K\right|$ has its maximum value;

For electrons:

$$\varepsilon_n = \frac{\max_m \left|n_m^{K+1} - n_m^K\right|}{max(C_0, n_m^K)} \qquad 14\text{-}7$$

For holes:

$$\varepsilon_p = \frac{\max_m \left|p_m^{K+1} - p_m^K\right|}{max(C_0, P_m^K)} \qquad 14\text{-}8$$

For lattice temperature:

$$\varepsilon_{T_L} = \frac{(T_L)_{nmax}^{K+1} - (T_L)_{nmax}^K}{(T_L)_{nmax}^{K+1}} \qquad 14\text{-}9$$

where nmax is the node where $(T_L)_i^{K+1}$ has its maximum value;

For carrier temperature:

$$\varepsilon_{T_c} = max(\varepsilon_{T_n}, \varepsilon_{T_p}) \qquad 14\text{-}10$$

where

$$\varepsilon_{T_n} = \frac{\max_m \left|(T_n)_m^{K+1} - (T_n)_m^K\right|}{\max_m (T_n)_m^{K+1}} \qquad 14\text{-}11$$

$$\varepsilon_{T_p} = \frac{\max_m \left| (T_p)_m^{K+1} - (T_p)_m^K \right|}{\max_m (T_p)_m^{K+1}} \quad 14\text{-}12$$

For drift diffusion, iterations are terminated if the following criteria are satisfied:

$$\varepsilon_v = \leq P_{tol}^x \quad 14\text{-}13$$

$$\varepsilon_n = \leq c_{tol}^x \quad 14\text{-}14$$

$$\varepsilon_v = \leq C_{tol}^x \quad 14\text{-}15$$

In nonisothermal drift diffusion, iterations are terminated if Equations 14-10 to 14-12 are satisfied, the current convergence criteria in Equations 14-4, 14-5 are met and:

$$\varepsilon_{T_L} < TL_{tol}^x \quad 14\text{-}16$$

In Gummel's method with energy balance equations NITGUMM iterations in which only the non-linear Poisson equation is solved, will always be done. Gummel's method with energy balance equations is terminated if 14-16 is fulfilled, the carrier temperature convergence criteria

$$\varepsilon_{T_c} \leq TC_{tol}^x \quad 14\text{-}17$$

is achieved, and one of the following conditions is valid:

a) NITGUMM < K+1 ≤ NITGUMM+NT1 and 14 -13 is fulfilled;

b) NT1 + NITGUMM < K+1 ≤ NITGUMM + NT3 and:

$$\varepsilon_n \leq P_{tol}^x \cdot w \quad m \quad 14\text{-}18$$

where w = 10

c) NT1 + NITGUMM < K + 1 ≤ NITGUMM + NT3, the current convergence criteria 14-4 and 14-5 are satisfied, and inequality 14-18 is valid for w = 100;

d) NT1 + NITGUMM < K+1 and 14-18 is valid for w = 100;

e) NT1 + NITGUMM < K+1 the current convergence criteria 14-4 and 14-5 are satisfied, and 14-18 is valid for w = 500.

The default values of the iteration parameters are NITGUMM = 5, NT0 = 4, NT1 = 10, NT3 = 100

Convergence Criteria For Newton's Algorithm

Relative update errors are defined as follows:

For potential

$$\varepsilon_v = \max_m \left| \varphi_m^{K+1} - \varphi_m^K \right| \qquad 14\text{-}19$$

For electron concentration:

$$\varepsilon_n = \frac{\max_m \left| n_m^{K+1} - n_m^K \right|}{max(C_0, n_m^K)} \qquad 14\text{-}20$$

For hole concentration:

$$\varepsilon_p = \frac{\max_m \left| p_m^{K+1} - p_m^K \right|}{max(C_0, p_m^K)} \qquad 14\text{-}21$$

For lattice temeprature and carrier temperature:

$$\varepsilon_{T_L} = \frac{\max_m \left| (T_L)_m^{K+1} - (T_L)_m^K \right|}{T_{scale}}, \qquad 14\text{-}22$$

$$\varepsilon_{T_n} = \frac{\max_m \left| (T_n)_m^{K+1} - (T_n)_m^K \right|}{T_{scale}}, \qquad 14\text{-}23$$

$$\varepsilon_{T_p} = \frac{\max_m \left| (T_p)_m^{K+1} - (T_p)_m^K \right|}{T_{scale}}, \qquad 14\text{-}24$$

where the scaling temperature Tscale is by default equal to 300K.

To define the RHS norms used in ATLAS, first represent the non-linear equations obtained after discretization at every node as

$$( ) \overrightarrow{(\chi)} = 0 \qquad 14\text{-}25$$

where α can be ψ, n, p, T_L, T_n and T_p for the potential equation, electron continuity equation, hole continuity equation, lattice temperature equation, electron temperature equation and hole temperature equation, respectively. $\vec{\chi}$ represents the vector of unknowns.

The RHS norms in ATLAS is then defined as follows:

For the potential equation:

$$\varepsilon_v^g = \max_m |F\varphi| \qquad 14\text{-}26$$

For the electron and continuity equations:

$$\varepsilon_n^g = \max_m |F_n| \cdot C_T \qquad 14\text{-}27$$

$$\varepsilon_p^g = \max_m |F_p| \cdot C_T \qquad 14\text{-}28$$

where CT = 10-4 $\sqrt[4]{N_c N_v}$ kT

For the lattice temperature and carrier temperature equations:

$$\varepsilon_{T_L}^g = \max_m |F_{T_L}| \qquad 14\text{-}29$$

$$\varepsilon_{T_n}^g = \max_m |F_{T_n}| . \qquad 14\text{-}30$$

$$\varepsilon_{T_p}^g = \max_m |F_{T_p}| \qquad 14\text{-}31$$

Newton iterations are terminated if one of the following criteria is satisfied:

a) the parameter XNORM is true and:

$$\varepsilon_v \le P_{tol}^x \cdot w, \qquad 14\text{-}32$$

$$\varepsilon_n \le c_{tol}^x \cdot w, \qquad 14\text{-}33$$

$$\varepsilon_p \le P_{tol}^x \cdot w, \qquad 14\text{-}34$$

$$\varepsilon_{T_L} \le TL_{tol}^x \cdot w, \qquad 14\text{-}35$$

$$\varepsilon_{T_n} \le TC_{tol}^x \cdot w, \qquad 14\text{-}36$$

$$\varepsilon_{T_p} \le TL_{tol}^x \cdot w, \qquad 14\text{-}37$$

where w = 1

b) the parameter RHSNORM is true, conditions for Equations 14-36, 14-37 are satisfied for w = 1 and:

$$\varepsilon_v^g \le P_{tol}^r \cdot w_1, \quad 14\text{-}38$$

$$\varepsilon_n^g \le C_{tol}^r \cdot w_1, \quad 14\text{-}39$$

$$\varepsilon_p^g \le P_{tol}^r \cdot w_1, \quad 14\text{-}40$$

$$\varepsilon_{T_L}^g \le TL_{tol}^r \cdot w_1, \quad 14\text{-}41$$

$$\varepsilon_{T_n}^g \le TC_{tol}^r \cdot w_1, \quad 14\text{-}42$$

$$\varepsilon_{T_p}^g \le TL_{tol}^r \cdot w_1, \quad 14\text{-}43$$

c) both the XNORM and RHSNORM parameters are true, the convergence criteria for Equations: 14-36 and 14-37 for carrier temperature are fulfilled, and for the inequalities (Equations 14-32 and 14-38), (Equations 14-33 and 14-39), (Equations 14-34 and 14-40), (Equations 14-35 and 14-41) one of the conditions is fulfilled for every pair.

d) if the current convergence criteria are satisfied and condition a) or condition b) or condition c) is fulfilled for: w = w_1 = WEAK

Convergence Criteria For Block Iteration

For the potential equation and the continuity equations, the X norms and RHS norms are defined as for the Newton method by Equations 14-19 to 14-21 and Equations 14-26 to 14-27.

With s as an index that denotes the number of block iterations, update errors between successive pairs of block iterations for lattice and carrier temperature are defined by the following expression:

For lattice temperature:

$$\varepsilon_{T_L}^{B_1} = \frac{\left|(T)L_{nmax}^{s+1} - (T_L)_{nmax}^{s}\right|}{\max_m (T_L)_m^{s+1}} \quad 14\text{-}44$$

where mmax is the number of the node where $(TL)_m$ has its maximum value.

For electron temperature:

$$\varepsilon_{T_n}^{B_1} = \frac{\max_m \left|(T_n)_m^{s+1} - (T_n)_m^{s}\right|}{\max_m (T_n)_m^{s+1}} \quad 14\text{-}45$$

For hole temperature:

$$\varepsilon_{T_p}^{B_1} = \frac{\max_m \left| (T_p)_m^{s+1} - (T_p)_m^{s} \right|}{\max_m (T_p)_m^{s+1}} \quad 14\text{-}46$$

Block iterations are terminated if :

$$\varepsilon_{T_L}^{B_1} \leq TL_{tol}^{x} \quad 14\text{-}47$$

$$\varepsilon_{T_n}^{B_1} \leq TC_{tol}^{x} \quad 14\text{-}48$$

$$\varepsilon_{T_p}^{B_1} \leq TL_{tol}^{x} \quad 14\text{-}49$$

and one of the following criteria is fulfilled:

1. parameter XNORM is true and Equations 14-32, 14-33 and 14-34 are valid for w =
2. parameter RHSNORM is true and Equations 14-38, 14-39, and 14-40 are valid for $w_1 = 1$
3. both XNORM and RHSNORM are true and for the pairs of inequalities (Equations 14-32 and 14-38), (Equations 14-33 and 14-39), (Equations 4-34 and 14-40) one of the conditions is satisfied for each pair.
4. the current convergence criteria are satisfied and condition 1 condition 2 or condition 3 is fulfilled for $w = w_1 =$ WEAK

Initial Guess Strategies

Non-linear iteration starts from an initial guess. The quality of the initial guess (i.e. how close it is to the final solution) affects how quickly the solution is obtained, and whether convergence is achieved. Users of ATLAS are not required to specify an initial guess strategy. If no strategy is defined, ATLAS follows certain rules that implement a sensible, although not necessarily optimum, strategy.

There is some interaction between the choice of non-linear iteration scheme and the initial guess strategy. Decoupled iteration usually converges linearly, although perhaps slowly, even from a relatively poor initial guess. Newton iteration converges much faster for a good initial guess, but fails to converge if started from a poor initial guess.

One very simple initial guess strategy is to use the most recent solution as the initial guess. Of course, there is no previous solution for the first calculation in a series of bias points. In this case, an intial solution is obtained for equilibrium conditions. There is no need to solve the current continuity equations at equilibrium, and a solution of Poisson's equation is quickly obtained.

It is also possible to modify the initial guess in a way that makes some allowance for the new bias conditions. Typical strategies include:

- Using two previous solutions and interpolation to project a new solution at each mesh point.
- Solving a form of current continuity equation with carrier concentrations held constant. This strategy yields an improved estimate of new potential distribution.
- Modifying the majority carrier quasi-Fermi levels by the same amount as the bias changes.

Parameters on the SOLVE statement can be used to specify an initial guess strategy. Five initial guess strategies are available.

- INITIAL starts from space charge neutrality throughout the device. This choice is normally used to calculate a solution with zero applied bias.
- PREVIOUS uses the currently loaded solution as the initial guess at the next bias point. The solution is modified by setting a different applied bias at the contacts.
- PROJECTION takes two previous solutions whose bias conditions differ at one contact and extrapolates a solution for a new applied bias at that contact. This method is often used when performing a voltage ramp.
- LOCAL sets the applied bias to the specified values, and changes the majority carrier quasi-Fermi levels in heavily doped regions to be equal to the bias applied to that region. This choice is effective with Gummel iteration, particularly in reverse bias. It is less effective with Newton iteration.
- MLOCAL starts from the currently loaded solution and solves a form of the total current continuity equation that provides an improved estimate of the new potential distribution. All other quantities remain unchanged. MLOCAL is more effective than LOCAL because it provides a smooth potential distribution in the vicinity of p-n junctions. It is usually more effective than PREVIOUS because MLOCAL provides a better estimate of potential. This is especially true for highly doped contact regions and resistor-like structures.

When a regrid is performed, the solution is interpolated from the original grid onto a finer grid. This provides an initial guess that can be used to start the solution of the same bias point on the new grid. Although the initial guess is an interpolation of an exact solution, this type of guess does not provide particularly fast convergence.

Recommendations And Defaults

It is not normally required for the user to specify *any* initial guess parameter.

The INITIAL parameter is normally used only to obtain a thermal equilibrium solution. The PREVIOUS, PROJECT, LOCAL, and MLOCAL parameters are used for other bias points. PREVIOUS and PROJECTION are the most frequently used. PROJECTION is normally prefered to PREVIOUS when it is available, i.e., when there are two previous solutions differing in the bias applied to the appropriate terminal. PREVIOUS is required for transient simulations, and for simulations that use current boundary conditions. LOCAL and MLOCAL tend to work well for reverse-biased devices, and are especially efficient when trying to increase very large voltage increments. By default, PROJECTION is used whenever two appropriate solutions are available. Otherwise, the PREVIOUS guess is used, unless there is no previous solution, in which case INITIAL is used.

The DC Curve-Tracer Algorithm

Tracing I-V curves for complicated device phenomena such as breakdown or latchup can be very difficult using conventional methods. ATLAS includes a special purpose DC curve-tracing algorithm that overcomes these problems. This algorithm is based on a dynamic load-line technique that adapts the boundary conditions for each step. The approach implemented in ATLAS for curve tracing is based on the work describe in Reference 118.

The key idea is that bias conditions can evolve smoothly between the limits of pure voltage control and pure current control. This is achieved using external resistors that adapt dynamically to the shape of the I-V curves to ensure that at each point the load line is perpendicular to the local tangent of the trace. With this value for the external resistor, the solution is projected to the next operating point by stepping the external voltage. Once the solution has converged, a new external resistance is calculated based on the new tangent information, and the process repeats itself.

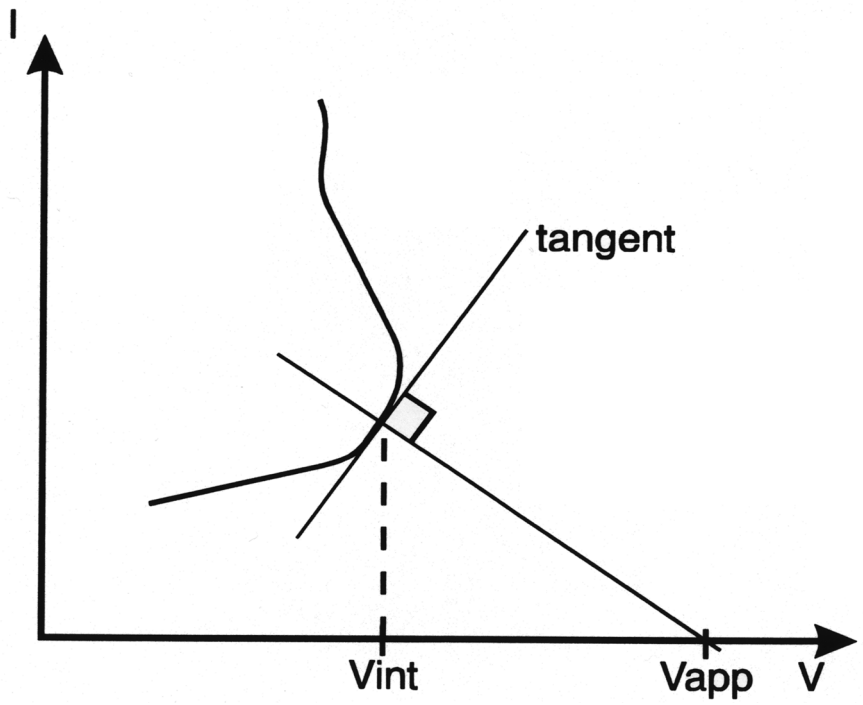


Figure 14-1: Load the algorithm used in the Curve Tracer

The curve tracing capability is activated by specifying the CURVETRACE parameter in the SOLVE statement. Prior to this, the CURVETRACE statement is used to set the parameters for the tracing algorithm. These parameters are the name CONTR.NAME of the ramped electrode (which will be referred to as a control electrode), the initial voltage increment STEP.INIT, the upper bound of the tracing curve, and additional parameters if they differ from the default values. The Upper bound parameter, END.VAL, is used to stop tracing. If the VOLT.CONT parameter is specified, END.VAL is a voltage. If the CURR.CONT parameter is specified, END.VAL is a current.

The applied voltage at each step is altered in accordance with the slope of the I-V curve. The resistor between the applied voltage and the semiconductor is also changed dynamically to ensure the voltage at the semiconductor (VINT) is smoothly varied along the I-V curve. If STEP.CONT is specified, the number of operational points on a trace will not exceed specified parameter STEPS.

Transient Simulation

When transient simulation is performed, the carrier continuity equations are integrated in the time domain. Time integration schemes differ in their accuracy, in the number of previous time levels they employ, and in their stability properties.

Accuracy is usually referred to as being ‘nth order’, where n is usually an integer between 1 and 4. In the limit of a small timestep, the magnitude of the local truncation error (LTE) introduced by the time integration scheme is proportional to the nth power of the timestep. Schemes that require the storage of solutions at timesteps previous to the most recent one are unattractive due to storage requirements. Single step integration schemes that use the solution at only one previous time level have a maximum order of 2.

The continuity equations are ‘stiff’, i.e. they are impacted by phenomena with very short timescales. Stiffness imposes stringent requirements on the stability of time integration schemes. Two forms of stability, A-stability and L-stability, are important. A-stability guarantees that errors introduced at one time step will not increase at the next timestep. L-stability guarantees that errors will decay even

for large time step values. A-stability is a requirement for any practical scheme. L-stability is extremely desirable to avoid non-physical 'ringing'.

Most device simulation codes use a simple first-order (implicit) backward difference formula for time integration. This scheme, which is known as BDF1, is both A-stable and L-stable. Unfortunately, the scheme is inaccurate for typical timesteps of interest. Second order accuracy is obtained using the the trapezoidal rule (TR) for time integration. This scheme is A-stable, but it is not L-stable. This means that solutions exhibit non-physical 'ringing' unless very small timesteps (much smaller than those dictated by LTE considerations) are used. The BDF2 scheme is second order, and both A-stable and L-stable, but it uses solutions from two previous time levels, and is also less accurate than TR.

For drift-diffusion calculations, ATLAS uses a composite TR-BDF2 scheme that was developed by Bank et. al.[46] This method is one-step, second order, and both A-stable and L-stable. An estimate of the LTE is obtained at each timestep, and this estimate is used to automatically adapt the timestep.

Different schemes are used for transient solutions that include lattice heating or energy balance. If lattice heating is included the block iterative procedure is organized at each time step in the same way as for the steady state case. If energy balance is selected, the absolutely stable half-implicit scheme[49] is used. Automatic timestep selection with local error control is implemented in this case. The tolerance may be specified using the TOL.TIME parameter in the METHOD statement.

Note: The user normally specifies only the initial timestep with the TSTEP parameter of the SOLVE statement. After this, time steps are derived from the LTE and will typically increase.

Small Signal and Large Signal Analysis

There are several ways to predict the small-signal and large-signal high-frequency properties of semiconductor devices (review of these different techniques was presented by Laux[46]). Frequency domain perturbation analysis (FPDA) can be used to determine the small-signal characteristics while Fourier analysis of transient responses (FATR) can be used for both small-signal and large signal response. Each technique has advantages and disadvantages.

Frequency Domain Perturbation Analysis

Frequency-domain perturbation analysis of a DC solution can be used to calculate small-signal characteristics at any user-specified frequency. The calculation proceeds in the following manner:

1. Variables are represented as the sum of the known DC component and a small unknown sinusoidal AC component.
2. All equations are expanded.
3. Differentiation in time becomes multiplication by the value of ω ($\omega = 2\pi$ frequency).
4. Products of AC quantities are neglected since they are small with respect to other quantities
5. The DC solution is subtracted.

What remains is a complex linear system whose unknowns are the AC components of the solution. Solving this linear system with appropriate boundary conditions yields small-signal characteristics.

The coefficient matrix of the complex linear system is simply the Jacobian associated with the DC operating point, with some terms on the leading diagonal supplemented by $j\omega$. The Jacobian is available 'for free' if the DC solution was calculated using Newton iteration. This is a very attractive feature of the Newton method. The resulting linear complex system can be solved by an iterative or a direct method. A form of successive over relaxation (SOR) works well for frequencies significantly below the cutoff frequency, but fails at higher frequencies. More sophisticated iterative techniques can

be used at higher frequencies, but it is simpler and more reliable to switch to direct sparse matrix methods if SOR fails to converge.

Frequency domain perturbation analysis is extremely attractive when the full Newton method is used to calculate a DC solution. The method works for all frequencies. It requires a predictable amount of computation which is quite low with respect to other calculations, since only a single linear system is solved.

Frequency domain perturbation analysis is invoked in ATLAS by specifying the appropriate parameters in the SOLVE statement. The AC.ANALYSIS parameter specifies that this analysis is to be performed. TERMINAL specifies the contact whose terminal voltage is to be perturbed. A full characterization requires that all but one of the device terminals is perturbed. FREQUENCY, FSTEP, and NFSTEPS determine the frequencies for which solutions are obtained. FREQUENCY specifies the initial frequency. FSTEP and NFSTEPS define a loop on frequencies. If MULT.FREQ is specified, the frequency is multiplied by FSTEP at each increment. This is useful for characterizing the small-signal response over several decades of frequency. The solution method can be specified using the SOR, DIRECT, and AUTO parameters. AUTO starts out using SOR but switches to DIRECT if convergence problems are encountered.

Fourier Analysis Of Transient Responses

FATR is a post-processing step which must be performed on a LOG file which contains transient data. The FOURIER statement performs a Fast Fourier Transform (FFT) on the time domain data transforming it into the frequency domain.

The syntax of the FOURIER statement is shown in Table 14.2.

Table 14.2. User-Specifiable Parameters for Equation 9-2

Statement	Parameter	Default
FOURIER	INFILE	
FOURIER	OUTFILE	
FOURIER	T.START	
FOURIER	T.STOP	
FOURIER	FUNDAMENTAL	
FOURIER	MAX.HARMONIC	
FOURIER	NUM.SAMPLES	64
FOURIER	INTERPOLATE	FALSE
FOURIER	COMPLEX.VALUES	FALSE

The explanation for the FOURIER parameters are as follows:

- INFILE – input log file. This should contain data from a transient simulation.
- OUTFILE – file output file for the Fourier transform.

The following parameters are optional:

- T.START – start of time data to be used for the FFT. The default value is the first time point in the input log file.

- T.STOP – end of time data to be used for the FFT. The default value is the last time point in the input log file.
- FUNDAMENTAL – fundamental frequency. If this is not specified, then the fundamental frequency is set to 1/(T.STOP - T.START). If the fundamental frequency is specified then T.STOP is set to T.START + 1/FUNDAMENTAL.
- MAX.HARMONIC – maximum harmonic frequency that the FFT should calculate. This will automatically calculate the correct number of samples (NUM.SAMPLES) required to generate this frequency. FUNDAMENTAL must be specified when MAX.HARMONIC is used.
- NUM.SAMPLES – number of samples. This should be an integer power of 2, i.e. 2^n, where n is a positive integer. The default value is 64 unless the MAX.HARMONIC parameter is specified. In this case, the number of samples is set to the nearest integer power of 2 which will generate this frequency.
- INTERPOLATE – performs linear interpolation on input data with non-uniform timesteps. This interpolates the data on to uniform timesteps. Interpolation of data can introduce addition (inaccurate) harmonic values into the FFT, which would not occur if uniform time is taken. INTERPOLATE must be used if the log file contains non-uniform time steps.
- COMPLEX.VALUES – prints the real and imaginary components to file, as well as the magnitude and phase.

Note: FFT works best with uniform time steps, therefore DT.MIN and DT.MAX on the METHOD statement should be set to the same value. The time step should be set to: time step = 1/(number of samples * fundamental frequency).

The FFT can then calculate harmonic frequencies up to:

(number of samples/2 - 1)*fundamental frequency.

Small-signal data at high frequencies can be obtained by calculating the terminal current responses to terminal voltage perturbations. The currents and voltages should then Fourier analyzed. Their ratio at each frequency provides admittance data for that frequency. The voltage perturbations are normally selected to have an analytic form with a known Fourier transform. Care must be taken to self-consistently account for geometric capacitances when step function voltage perturbations are used. The advantages of this technique are that it can be used whenever transient calculations are possible, and each transient solution gives information over a broad range of frequencies. The main disadvantage is that transients become very long when low frequency effects are investigated.

Overall Recommendations

1. Use frequency-domain perturbation analysis when it is available. The method works for arbitrary frequencies and does not require transient calculations.
2. Use Fourier analysis of transient responses for high frequencies when FPDA is not available but transient calculations are possible.

Differences Between 2D and 3D Numerics

With respect to numerical techniques there are several differences between 2D and 3D simulations.

First, with respect to the nonlinear iteration strategies, all three strategies, NEWTON, GUMMEL and BLOCK are supported in 2D simulation, whereas, only NEWTON and GUMMEL are supported for 3D simulations. Implementation of the BLOCK strategy is expected in a future release.

Second, solution of the linear subproblem is handled differently for 2D and 3D simulations. As previously noted the computational burden of solving the linear subproblem increases with the size of the solution domain. For smaller problems direct methods are quicker while for larger problems iterative methods are preferred. It turns out that the point at which the iterative methods become less burdensome roughly coincides with the transition between 2D and 3D domains. As such the default method for 2D simulations is a direct solver. For 3D simulations, the default method is an iterative solver. By default ILUCGS is applied to 3D simulations. ILUCGS is an acronym for incomplete lower upper decomposition conjugate gradient squared. Two alternative iterative solvers are also available for 3D simulations. BICGST (BICGST on the METHOD statement) is an acronym for biconjugate gradient squared stabilized. GMRES (GMRES on the METHOD statement) is an acronym for generalized minimum residual. Direct methods can be used for 3D simulation by specifying DIRECT on the METHOD statement. Practical experience shows that for 3D simulations either of the iterative methods are faster than the direct method. However, in some cases the accuracy produced by the iterative methods can prevent convergence in the nonlinear outer loop. For 2D simulations only direct methods are supported.

The DC curve tracer algorithm and small signal AC analysis are also not currently supported for 3D simulations.

[This page intentionally left blank]

Chapter 15: Statements

Overview

Input Language

This chapter contains a complete description (in alphabetical order) of every statement and parameter used by any of the ATLAS products, except for MIXEDMODE. (MIXEDMODE specific parameters are documented in the MIXEDMODE Chapter). The following documentation is provided for each statement:

- The statement name
- The product for which the statement is applicable
- The syntax of the statement
- A list of all statement parameters, their type, default value, and units
- A description of each parameter
- An example of the correct usage of each statement

Note: An error message will be generated if you attempt to specify a statement for a simulator which you have not purchased. For example, the BEAM statement may only be used if you have purchased LUMINOUS or LUMINOUS3D.

Syntax Rules

In this manual, we refer to *statements*, and *parameters*. An input deck line is referred to as a statement (or statement line). Since statements and parameters are not case sensitive, they may be entered using either uppercase or lowercase letters.

A statement is specified in the general format

```
<STATEMENT> <PARAMETER>=<VALUE>
```

where STATEMENT is the statement name, PARAMETER is the parameter name, and VALUE is the parameter value. The space character is used to separate the parameters in a statement.

The words and numbers which follow a statement are parameters of that statement. A *word* is an alphanumeric string which is terminated either by a space or by a carriage return. A *number* is a numeric or alphanumeric string which is terminated either by a space or by a carriage return. Numerical values may range from 10^{-38} to 10^{38}. A number may contain the symbols + (positive), - (negative), and/or E (decimal notation). For example:

```
+10 -1.234E5 .003 -.12E+10
```

Four types of parameters are used by the ATLAS products real, integer, logical, and character. These parameter types are explained in Table 15-1.

Table 15.1. Types of Parameters

Parameter	Description	Value Req	Example
Character	Any character string	Yes	INFILE=NMOS.DOP
Integer	Any whole number	Yes	REGION=2
Logical	A true or false condition	No	GAUSSIAN
Real	Any real number	Yes	X.MIN=0.52

Any parameter which does not have a logical value must be specified in the form PARAM=VAL where PARAM is the name of the parameter and VAL is the value of the parameter. Logical parameters must be separated from other parameters or commands by a space.

For example, in the statement

```
DOPING UNIFORM CONCENTRATION=1E16 P.TYPE
```

parameters UNIFORM and P.TYPE have logical values and the CONCENTRATION parameter has a value of $1x10^{16}$ (real).

Logical parameters can be turned off (switched from true to false) by placing a caret (^) in front of the logical parameter. For example, in the statement

```
DOPING UNIFORM CONCENTRATION=1E16 ^P.TYPE
```

the parameter P.TYPE has been set to false.

Mnemonics

It is not always necessary to input the entire statement or parameter name. ATLAS requires only that you input enough letters to distinguish that command or parameter from other commands or parameters. For example, DOP may be used to abbreviate the DOPING command.Excessive truncation is not recommended since future ATLAS syntax might make short abbreviations become ambiguous.

Continuation Lines

Since it may be necessary for a statement line to contain more than 256 characters, ATLAS allows the user to specify continuation lines. To continue a line put a backslash, (\) character at the end of the line that is to be continued. When ATLAS encounters the backslash, it will interpret the next line to be a continuation of the current line. The PISCES-II continuation of using a (+) at the start of the subsequent line is not supported in ATLAS.

Comments

Comments are indicated either by COMMENT command, or by a pound sign (#). All characters on a line which follow a comment indicator (COMMENT or #) will not be analyzed by ATLAS.

Synonyms

Some parameters have synonyms — parameters which have a different name but the same functionality. A parameter's synonym, if it exists, is listed in the parameter descriptions for each statement.

Pseudonyms

Throughout the statement descriptions in the remainder of this chapter, pseudonyms are used either to indicate a group of parameters or to indicate the value of a particular parameter. A < symbol indicates the start of a pseudonym(s). A > symbol indicates the end of a pseudonym(s). Pseudonyms will be separated from one another by a space character ( ).

For example, LOC might indicate a group of location parameters and FILENAME might indicate the name of a file which must be specified.

Symbols

The following symbols are used in the statement descriptions throughout the rest of this chapter.

< Indicates the start of a list of pseudonyms.

> Indicates the end of a list of pseudonyms.

| Separates parameters or pseudonyms which are mutually exclusive. Only one of these parameters may be used in a statement.

[Indicates the start of an optional command, parameter, or pseudonym.

] Indicates the end of an optional command, parameter, or pseudonym.

Expressions

ATLAS does not support arithmetic expressions in the syntax. Expressions can be evaluated and used using the SET or EXTRACT statements however.

BEAM

BEAM — specifies an optical input signal in the form of a collimated beam of light. This statement is used with LUMINOUS or LUMINOUS3D.

Syntax

```
BEAM <parameters>
```

Parameter	Type	Default	Units
ANGLE	Real	0.0	Degrees
BACK.REFL	Logical	False	
DEVICE	Character		
F.OXGENERATE	Character		
F.RADIATE	Character		
F.RADIATE3D	Character		
F.REFLECT	Character		
FRONT.REFL	Logical	False	
ITERATION	Integer	20	
LENS.INDEX	Real	1.0	
LENS.PLANE	Real	0.0	μm
LENS.RADIUS	Real	0.0	μm
LENS.X	Real	0.0	μm
LENS.Y	Real	0.0	μm
LENS.Z	Real	0.0	μm
MAX.WINDOW	Real	$1.0x10^{20}$	μm
METAL.REFLECT	Logical	False	
MIN.POWER	Real	0.0	μm
MIN.WINDOW	Real	$-1.0x10^{20}$	μm
NUMBER	Integer	1	
NX	Interger	10	
NZ	Integer	10	
PERIODIC	Logical	False	
PHI	Real	0.0	Degrees
POLARIZE	Real	0.0	Degrees
POWER.FILE	Character		

Parameter	Type	Default	Units
POWER.SCAL	Real	1.0	
QUANTUM.EFF	Real	1.0	
RAYS	Integer	1	
RAYTRACE	Character		
REFLECTS	Integer	1	
REL.POWER	Real	1.0	
THETA	Real	0.0	Degrees
THINEST	Real	0.0	µm
WAVELENGTH	Real	0.623	µm
WAVEL.END	Real	0.0	µm
WAVEL.NUM	Integer	1	
WAVEL.SCAL	Real	1.0	
WAVEL.START	Real	0.0	µm
X.ORIGIN	Real	0.0	µm
XMAX	Real	$-1.0x10^{20}$	µm
XMIN	Real	$1.0x10^{20}$	µm
Y.ORIGIN	Real	0.0	µm
Z.ORIGIN	Real	0.0	µm
ZMAX	Real	$-1.0x10^{20}$	µm
ZMIN	Real	$1.0x10^{20}$	µm

Description

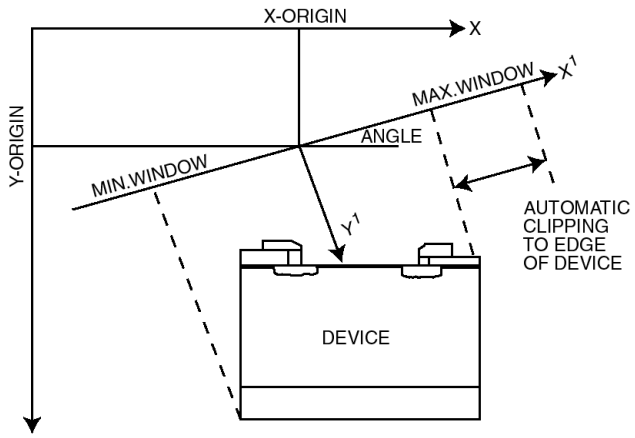


Figure 15-1: LUMINOUS Optical Source Coordinate System

ANGLE is the angle of propogation of the optical beam (see Figure 15-1 and 11-1). `ANGLE=90` is vertical illumination from the top of the device.

Synonym: `PHI`

BACK.REFL specifies that back side reflections are to be taken into account. When `BACK.REFL` is specified, the area outside the device domain is assumed to be a vacuum (ie. n = 1.0, k = 0.0)

DEVICE is used to specify the name of the device that the beam is associated with in MIXEDMODE simulation.

F.REFLECT specifies the name of a file containing a C-INTERPRETER function for specifying reflection coefficient models as a function of wavelength, position, and angle incidence (LUMINOUS only).

F.RADIATE specifies the name of a file containing a C-INTERPRETER function for specifying generation rate as a function of position and optionally time. This function can be used to simulate single event upset (LUMINOUS only).

F3.RADIATE is the same as the `F.RADIATE` parameter but is applied in 3-D. This is typicallyused for single event or photogeneration simulations with LUMINOUS3D.

FRONT.REFL specifies that front side reflections are to be taken into account. When `FRONT.REFL` is specified, the area outside the device domain is assumed to be a vacuum (ie. n=1.0, k= 0.0).

LENS.INDEX sepcifies the index of refraction of a lenslet (LUMINOUS3D only).

LENS.PLANE sepcifies specifies the minimum y-cordinate of the lenslet sphere (LUMINOUS3D only).

LENS.RADIUS sepcifies the radius of the spere defining a lenslet (LUMINOUS3D only).

LENS.X sepcifies the x-coordinate of the center of the spere defining a lenslet (LUMINOUS3D only).

LENS.Y sepcifies the y-coordinate of the center of the spere defining a lenslet (LUMINOUS3D only).

LENS.Z sepcifies the z-coordinate of the center of the spere defining a lenslet (LUMINOUS3D only).

MAX.WINDOW specifies the maximum x-value of the illumination window relative to the coordinate system of the optical beam (see Figure 15-1). The illumination window is always clipped to the device domain.

Synonym: `XMAX`

METAL.REFLECT sepcifies that all metals are to be treated as perfect reflectors (LUMINOUS3D only).

MIN.WINDOW specifies the minimum x-value of the illumination window relative to the coordinate system of the optical beam.

Synonym: `XMIN`

NUMBER specifies the beam number (from 1 to 10). This number is used by the SOLVE statement to specify the relative intensity of different beams. You may specify beam numbers in any order that you desire.

NX specifies the number of rays traced along the source beam's x axis for LUMINOUS3D (see fig11-3).

NZ specifies the number of rays traced along the source beam's z axis for LUMINOUS3D (see fig11-3).

PERIODIC specifies that for ray tracing, the structure is to be treated as periodic in the x and z directions. Rays exiting the sides of the device are wrapped around to the other side of the device (LUMINOUS3D only).

POLARIZE specifies the polarization of the optical beam at the origin. The polarization angle is the angle between the E vector and the device plane.

POWER.SCALE specifies a scale factor. This factor is multiplied by each of the relative powers in the spectrum file when multi-spectral simulations are performed. The POWER.SCALE parameter can be used to perform unit conversions.

POWER.FILE specifies the filename of a spectrum file. Spectrum file must be in the format

```
number of pairs
wavelength_1  power_1
wavelength_2  power_2
    . . .       . . .
    . . .       . . .
wavelength_n  power_n
```

Note: The power file must contain at least two power/wavelengths pairs.

QUANTUM.EFF is a quantum efficiency factor which specifies the number of carrier pairs generated per photon absorbed.

RAYTRACE specifies the name of a file where the results of a ray trace are saved. The ray trace may be viewed using TONYPLOT3D (LUMINOUS3D only)

RAYS specifies the number of rays the user wishes to split the optical beam into. LUMINOUS will automatically split the beam into enough rays to resolve the geometry. Use of the number parameter will cause further splitting of the optical beam (LUMINOUS only)

REFLECTS specifies the number of reflections that will be traced. When the value of the REFLECTS parameter is increased, the total number of rays traced increases non-linearly. We recommend that this parameter be used judiciously. For example, a single ray incident on three material layers will produce 4 rays if REFLECTS=0 is specified, 10 rays if REFLECTS=1 is specified, and 24 rays if REFLECTS=2 is specified.

REL.POWER specifies the relative power in the beam when mono-spectral simulations are performed. This factor is multiplied by the power parameters specified in the SOLVE statement to give the total optical power in the beam.

THETA specifies the angle of rotation for the source beam direction of propogation relative to the x-y plane (see Fig. 11-2) (LUMINOUS3D only).

THINEST specifies the width of the thinnest ray to be traced (LUMINOUS only).

MIN.POWER specifies the minimum intensity relative to the source that a given ray will be traced. This is useful for limiting the numbers of rays traced.

WAVELENGTH specifies the optical wavelength of the source beam (in the vacuum) for mono-spectral simulations.

WAVEL.END specifies the maximum wavelength of the source beam (in the vacuum) when multi-spectral simulations are performed.

WAVEL.NUM specifies the number of wavelengths which will be used when multi-spectral simulations are performed.Spectral illumination is selected when the WAVEL.NUM parameter is greater than 1. Once multi-spectral simulations are selected, you must specify the POWER.FILE, WAVEL.START, and WAVEL.END parameters.

WAVEL.SCAL specifies the scale factor for the wavelengths which are used in multi-spectral simulations. Each of the wavelengths in the spectrum file is multiplied by this scale factor.

WAVEL.START specifies the minimum wavelength of the source beam (in the vacuum) when multi-spectral simulations are performed.

XMAX specifies the maximum x-coordinate in the source beam coordinate system for ray tracing in LUMINOUS3D (see figure 11-3).

XMIN specifies the minimum x-coordinate in the source beam coordinate system for ray tracing in LUMINOUS3D (see figure 11-3).

X.ORIGIN specifies the x-coordinate of the optical beam origin (see Figures 15-1 and 11-1). The beam must originate outside all device regions.

Y.ORIGIN specifies the y-coordinate of the optical beam origin (see Figures 15-1 and 11-1). The beam must originate outside all device regions.

ZMAX specifies the maximum z-coordinate in the source beam coordinate system for ray tracing in LUMINOUS3D (see figure 11-1).

ZMIN specifies the minimum z-coordinate in the source beam coordinate system for ray tracing in LUMINOUS3D (see figure 11-1).

Z.ORIGIN specifies the z-coordinate of the optical beam origin (see Figure 11-1). The beam must originate outside all device regions (LUMINOUS3D only).

Monochromatic Beam Example

This beam has a monochromatic spectrum with a wavelength of 0.6 μm. The beam originates at x=0.5 and y=-2.0. It has a 90 degree propogation angle and a beam width of 0.2 μm which is centered at the beam origin.During the ray trace calculation the rays will be terminated when the power level along the ray falls to 5% of the original power.

```
BEAM NUM=1 WAVELENGTH=0.6 X=0.5 Y=-2.0 ANG=90.0 MIN=-0.1 MAX=0.1 \
           MIN.POWER=0.05
```

Multispectral Beam Example

A multispectral beam (at a 45 degree angle) which originates at x=0.0 and y=-1.0. The multi-spectral source is imported from the spectrum file "source.spc". The spectrum is discretized into our wavelengths between 0.4 μm and 0.6 μm.

```
BEAM NUM=2 X=0.0 Y=-1.0 ANG=45.0 \
     POWER.FILE=SOURCE.SPC WAVEL.START=0.4 \
     WAVEL.END=0.6 WAVEL.NUM=4
```

LUMINOUS3D Lens Example

```
BEAM NUM=1 X.ORIGIN=2.5 Y.ORIGIN=-1.0 Z.ORIGIN = 2.5 ANG=90.0 WAVEL=0.6 \
           NX=10 NZ=10 LENS.X=2.5 LENS.Y=-0.5 LENS.Z=2.5 \
           LENS.INDEX=2.03 LENS.RADIUS=0.25 LENS.PLANE=-0.5
```

COMMENT,

COMMENT – allows comments to be placed in an ATLAS input file. ATLAS will print and display comment lines.

Syntax

```
COMMENT [<string>]
# [<string>]
```

string is any alphabetic, numeric, or alphanumeric sequence of characters.

Synonyms: #

Example

```
COMMENT  ATLAS is a copyright of Silvaco International
#        ATLAS is a copyright of Silvaco International
```

Note: The $ was allowed as a comment character in previous versions of ATLAS. This should be avoided and replaced by the # or COMMENT statement.

CONTACT

CONTACT — specifies the physical attributes of an electrode.

Note: If the CONTACT statement is not used for a given electrode, the electrode is assumed to be charge-neutral (ohmic).

Syntax

```
CONTACT NUMBER=<n>|NAME=<ename>|ALL [<wfp>] [<bc>] [<lcr>] [<link>]
```

Parameter	Type	Default	Units
ALL	Logical	False	
ALPHA	Real	0	cm
ALUMINUM	Logical	False	
BARRIER	Logical	False	
CAPACITANCE	Real	0	F/μm
COMMOM	Character		
CON.RESIST	Real	0	$\Omega \cdot cm^2$
CURRENT	Logical	False	
DEVICE	Character		
E.TUNNEL	Logical	False	
ELE.CAP	Integer		
F.ETUNNEL	Character		
FACTOR	Real	0	
FLOATING	Logical	False	
FG.CAP	Real	0.0	F/μm
INDUCTANCE	Real	0	H·μm
ME.TUNNEL	Real	1.0	
MO.DISILICIDE	Logical	False	
MOLYBDENUM	Logical	False	
MULT	Logical	False	
NAME	Character		
NEUTRAL	Logical	True	
N.POLYSILICON	Logical	False	

Parameter	Type	Default	Units
NUMBER	Integer		
P.POLYSILICON	Logical	False	
RESISTANCE	Real	0	Ω.µm
SHORT	Logical	False	
SURF.REC	Logical	False	
TU.DISILICIDE	Logical	False	
TUNGSTEN	Logical	False	
VSURFN	Real	see description	cm/s
VSURFP	Real	see description	cm/s
WORKFUN	Real	0	V

Description

NAME specifies the name of a previously defined electrode. See ELECTRODE statement.

NUMBER specifies the contact number to be defined. It must be the number of a previously defined electrode. It is recommended that electrode names be used rather than numbers.

ALL defines the same properties for all electrodes.

DEVICE specifies which device the CONTACT statement applies to in MIXEDMODE.

wfp is one of the work function parameters described below. It is permitted to either specify the name of a material or a work function value (WORKFUN parameter).

bc is one or more of the boundary condition parameters .

lcr is one or more of the external parasitic element parameters .

link is one or more of a set of parameters that allow the user to associate two or more electrodes electrically.

Workfunction Parameters

ALUMINUM specifies aluminum as the contact material for the electrode. This sets the workfunction to 4.10V. Note that this parameter should not be set if an ohmic contact is required.

MOLYBDENUM specifies molybdenum as the contact material for the electrode. This sets the work function of the electrode to 4.53V.

MO.DISILICIDE specifies molybdenum disilicide as the contact material for the electrode. This sets the work function of the electrode to 4.80V.

NEUTRAL specifies that the electrode is ohmic. This is the deafult characteristic of an electrode.

N.POLYSILICON specifies n+ doped polysilicon as the contact material for the electrode. This sets the work function to 4.17V.

P.POLYSILICON specifies p+ polysilicon as the contact material for the electrode. This sets the work function to 4.17V + Eg(Si).

TU.DISILICIDE specifies tungsten disilicide as the contact material for the electrode. This sets the work function to 4.80V.

TUNGSTEN specifies tungsten as the contact material for the electrode. This sets the work function to 4.63V.

WORKFUN specifies the work function of the electrode in V. This parameter must be specified in the form WORKFUN=**n** where **n** is a real number. This specification is absolute workfunction and not workfunction difference to the semiconductor.

Note: If no WORKFUN or material type parameter is specified the electrode is assumed to be an ohmic contact

---

Boundary Conditions

CURRENT specifies current boundary conditions. If specified, CAPACITANCE, CON.RESIST, INDUCTANCE or RESISTANCE may not be specified.

FLOATING specifies a charge boundary condition. This parameter is used to specify the floating gate in EPROM devices. This parameter can only be used for insulated contacts. If specified, CAPACITANCE, CON.RESIST, INDUCTANCE or RESISTANCE may not be specified, but special syntax exists for adding capacitances to a floating contact. See EL<n>.CAP and FG<n>.CAP

Synonym: CHARGE

ALPHA specifies the linear, dipole lowering coefficient. This parameter has no effect unless the BARRIER parameter has been specified.

BARRIER turns on the barrier lowering mechanism for Schottky contacts.

E.TUNNEL specifies that the Schottky tunnelling model for electrons will be used. E.TUNNEL will also enable the SURF.REC boundary condition which models the thermionic emission in a Schottky contact.

F.ETUNNEL specifies the name of a file containing a C-INTERPRETER function that specifies electron tunnelling at a Schottky contact.

ME.TUNNEL specifies the relative effective mass for use in the electron tunneling mode (see E.TUNNEL).

SURF.REC specifies that finite surface recombination velocities are used at the respective contact. This parameter must be specified in the form SURF.REC [VSURFN=<n>] [VSURFP=<p>], where **n** and **p** are real numbers.

VSURFN specifies the actual surface recombination velocities for electrons (V_{sn}). If this parameter is not specified, its default value is calculated by Equation 15-1.

$$V_{sn} = \frac{\mathrm{ARICHN}T^2}{qN_C} \qquad 15\text{-}1$$

where ARICHN is the effective Richardson constant for electrons. This constant accounts for quantum mechanical reflections and tunneling.

VSURFP specifies the actual surface recombination velocities for holes (V_{sp}). If this parameter is not specified, its default value is calculated by Equation 15-2.

$$V_{sp} = \frac{\mathrm{ARICHP}T^2}{qN_V} \qquad 15\text{-}2$$

where ARICHP is the effective Richardson constants for holes. This constant accounts for quantum mechanical reflections and tunneling.

Contact Parasitics

Note: There are restrictions on the allowed numerical methods and types of analysis possible when any form of parasitic element is attached to a contact. See the Getting Started section for details.

RESISTANCE specifies a lumped resistance value. You may not specify both RESISTANCE and CON.RESIST.

INDUCTANCE specifies an external inductance which is related to the specified electrode. A synonym is L

CAPACITANCE specifies a lumped capacitance value to be attached to the contact.

CON.RESIST specifies a distributed contact resistance. You cannot specify both CON.RESIST and RESIST.

Electrode Linking Parameters

These parameters allow one electrode to be biased as a function of another electrode. This allows separate regions of the same physical contact to be linked together. For example, in the statement:

```
CONTACT NAME=MYDRAIN COMMON=DRAIN FACTOR=4.6
```

Electrode ‘mydrain’ is linked to the electrode ‘drain’. The bias on ‘mydrain’ will always be equal to the bias on the drain plus 4.6. If the optional MULT parameter had been specified, the bias on my drain would be equal to the bias on the drain multiplied by 4.6.

COMMON specifies the electrode name to which the contact refered to by NAME is linked. Although the electrodes are linked separate currents will be saved for both electrodes unless SHORT is also specified. The electrode refered to in NAME should not appear on any SOLVE statments since it’s bias is now determined as a function of the electrode refered to by COMMON.

SHORT specifies that the electrode referred to by NAME is shorted to the electrode specified by the COMMON parameter. This implies that the two electrodes will be treated as one and only one value will be written to log files and in the run time output.

FACTOR specifies the constant offset voltage (or current) between the electrodes refered by by NAME and COMMON. By default, FACTOR is added the defined voltage.

MULT specifies that FACTOR is a multiplier.

Floating Gate Capacitance Parameters

In same cases it is desirable to simulate floating gate structures, in 2D, which have control gates that are longer in the unsimulated dimension than the floating gate. In these cases the user can specify the following parameters to account for addition capacitance between the floating gate and the control gate and other electrodes.Up to four extra capacitances are allowed so in the following **<n>** is an integer number between 1 and 4.

EL<n>.CAP specifies the name of the electrode to which the extra capacitance is linked.

FG<n>.CAP specifies the additional capacitance per unit length to be added between the floating gate and electrode specified in EL<n>.CAP.

Schottky Barrier and Surface Recombination Example

This example defines all electrodes except number 2 (aluminum) to be neutral. Electrode number 2 also includes finite surface recombination velocities and barrier lowering. A definition in the second statement overrides that definition in the first statement.

```
CONTACT ALL NEUTRAL
CONTACT NUMBER=2 ALUMINUM SURF.REC BARRIER
```

Parasitic Resistance Example

This example attaches a lumped resistor with a value of $10^5 \Omega\mu m$ to the substrate. A distributed contact resistance of $10^{-6} \Omega \cdot cm^2$ is included on the drain.

```
CONTACT NAME=substrate RESISTANCE=1E5
CONTACT NAME=drain CON.RESIST=1E-6
```

Floating Gate Example

This syntax defines a floating contact with a workfunction equal to 4.17eV. An extra 1fF/um capacitance is added between this electrode and the electrode named 'cgate'

```
CONTACT NAME=fgate FLOATING N.POLY EL1.CAP=cgate FG1.CAP=1e-15
```

Note: The command, `MODELS PRINT`, can be used to echo the back `contact` workfunction and parasitic elements settings to the run-time output

---

CURVETRACE

CURVETRACE – sets the parameters for the automatic curve tracing routine.

Syntax

```
CURVETRACE <params>
```

Parameter	Type	Default	Units
ANGLE1	Real	5	
ANGLE2	Real	10	
ANGLE3	Real	15	
BEG.VAL	Real	0.0	V
CONTR.ELEC	Integer		
CONTR.NAME	Character		
CURR.CONT	Logical		
END.VAL	Real		
MINCUR	Real		
MINDL	Real	0.1	
NEXTST.RATIO	Real	2.0	
STEP.CONT	Logical		
STEP.INIT	Real	0.1	V
STEPS	Real	0.1	
TURNINGPOINT	Logical	False	
VOLT.CONT	Logical	False	

Note: The curvetracer functionality of ATLAS creates several temporary files in the local directory in which it is run. Two simultaneous ATLAS runs using curvetrace in the same directory will interfere with each other.

Description

ANGLE1, ANGLE2 and **ANGLE3** are critical angles (in degrees) affecting the smoothness and step size of the trace. If the difference in slopes of the last two solution points is less than ANGLE1, the step size will be increased for the next projected solution. If the difference lies between ANGLE1 and ANGLE2, the step size remains the same. If the difference is greater than ANGLE2, the step size is reduced. ANGLE3 is the maximum difference allowed, unless overridden by the MINDL parameter. ANGLE2 should always be greater than ANGLE1 and less than ANGLE3.

BEG.VAL is the value of the voltage at the starting point of the curve trace for the controlling electrode.

CONTR.ELEC is the number of the electrode that is designated as a control electrode.

CONTR.**NAME** is the name of the control electrode.

CURR.**CONT** denotes that a maximum current on the control electrode, specified by END.VAL is used as the upper bound on the trace.

END.**VAL** is used to stop tracing if the voltage or current of control electrode equals or exceeds END.VAL.

NEXTST.**RATIO** specifies which factor to use to increase the voltage step on the smooth parts of the I-V curve.

MINCUR may be used to set a small current value in order to switch from internal control electrode bias ramping to external ramping with load resistor. This parameter is recommended for small current breakdown simulation.

MINDL is the minimum normalized step size allowed in the trace. Usually the user does not need to adjust this parameter. Increasing MINDL will reduce the smoothness of the trace by overriding the angle criteria, resulting in more aggressive projection and fewer simulation points. Reducing MINDL will enhance the smoothness and increase the number of points in the trace.

STEPS is the number of operational points on a trace if STEP.CONT was specified.

STEP.**CONT** specifies that the trace will proceed for a certain number of simulation points.

STEP.**INIT** specifies initial voltage step size.

Note: To set a sweep of increasingly negative voltage in CURVETRACE it is only necessary to set STEP.INIT to be negative. Since all parameters are multiplier of STEP.INIT the whole voltage sweep will be negative.

TURNINGPOINT specifies that binary output solution files will be saved whenever the slope of the IV curve changes sign; i.e. there is a turning point. The name of the output file is "*soln.num*", where *num* is the number of the current solution.

VOLT.**CONT** denotes that a maximum voltage on the control electrode, specified by END.VAL is used as the upper bound on the trace.

Diode Breakdown Example

To trace a diode breakdown curve using current value as a termination criteria, the following statement may be used:

```
CURVETRACE CURR.CONT END.VAL=0.01 CONTR.NAME=anode \
           MINCUR=5E-12 NEXTST.RATIO=1.1  STEP.INIT=0.1
SOLVE CURVETRACE
```

DEFECT

DEFECT — activates the band gap defect model and sets the parameter values. This model can be used when thin-film transistor simulations are performed using the TFT product.

Syntax

```
DEFECTS [<parameters>]
```

Parameter	Type	Default	Units
AFILE	Character		
CONTINUOUS	Logical	True	
DEVICE	Character		
DFILE	Characater		
EGA	Real	0.4	eV
EGD	Real	0.4	eV
F.TFTACC	Character		
F.TFTDON	Character		
NGA	Real	$5.0x10^{17}$	cm^{-3}
NGD	Real	$1.5x10^{18}$	cm^{-3}
NTA	Real	$1.12x10^{21}$	cm^{-3}/eV
NTD	Real	$4.0x10^{20}$	cm^{-3}/eV
NUMBER	Real	All	
NUMA	Real	12	
NUMD	Real	12	
SIGGAE	Real	$1.0x10^{-16}$	cm^2
SIGGAH	Real	$1.0x10^{-14}$	cm^2
SIGGDE	Real	$1.0x10^{-14}$	cm^2
SIGGDH	Real	$1.0x10^{-16}$	cm^2
SIGTAE	Real	$1.0x10^{-16}$	cm^2
SIGTAH	Real	$1.0x10^{-14}$	cm^2
SIGTDE	Real	$1.0x10^{-14}$	cm^2
SOGTDH	Real	$1.0x10^{-16}$	cm^2
WGA	Real	0.1	eV

Parameter	Type	Default	Units
WGD	Real	0.1	eV
WTA	Real	0.025	eV
WTD	Real	0.05	ev

Description

The DEFECTS statement is used to describe the density of defect states in the band gap. Up to four distributions can be specified — two for donor-like states and two for acceptor-like states. Each type of state may contain one exponential (tail) distribution and one Gaussian distribution.

AFILE specifies the file name in which the acceptor state density distribution as a function of energy will be stored. This file can be examined using TONYPLOT.

CONTINUOUS specifies that the continuous defect integral model will be used.

DEVICE specifies which device the statement applies in mixed mode simulation

DFILE specifies the file name in which the donor state density distribution as a function of energy will be stored. This file can be examined using TONYPLOT.

EGA specifies the energy that corresponds to the Gaussian distribution peak for acceptor-like states. This energy is measured from the conduction band edge.

EGD specifies the energy that corresponds to the Gaussian distribution peak for donor-like states. This energy is measured from the valence band edge.

F.TFTACC specifies the name of a file containing a C-INTERPRETER function describing the distribution of acceptor state densities as a function of energy.

F.TFTDON specifies the name of a file containing a C-INTERPRETER function describing the distribution of donor state densities as a function of energy.

NGA specifies the total density of acceptor-like states in a Gaussian distribution.

NGD specifies the total density of donor-like states in a Gaussian distribution.

NTA specifies the density of acceptor-like states in the tail distribution at the conduction band edge.

NTD specifies the density of donor-like states in the tail distribution at the valence band edge.

NUMBER or **REGION** specifies the region index to which the DEFECT statement applies.

NUMA specifies the number of discrete levels that will be used to simulate the continuous distribution of acceptor states.

NUMD specifies the number of discrete levels that will be used to simulate the continuous distribution of donor states.

SIGGAE specifies the capture cross-section for electrons in a Gaussian distribution of acceptor-like states.

SIGGAH specifies the capture cross-section for holes in a Gaussian distribution of acceptor-like states.

SIGGDE specifies the capture cross-section for electrons in a Gaussian distribution of donor-like states.

SIGGDH specifies the capture cross-section for holes in a Gaussian distribution of donor-like states.

SIGTAE specifies the capture cross-section for electrons in a tail distribution of acceptor-like states.

SIGTAH specifies the capture cross-section for holes in a tail distribution of acceptor-like states.

SIGTDE specifies the capture cross-section for electrons in a tail distribution of donor-like states.

SIGTDH specifies the capture cross-section for holes in a tail distribution of donor-like states.

WGA specifies the characteristic decay energy for a Gaussian distribution of acceptor-like states.

WGD specifies the characteristic decay energy for a Gaussian distribution of donor-like states.

WTA specifies the characteristic decay energy for the tail distribution of acceptor-like states.

WTD specifies the characteristic decay energy for the tail distribution of donor-like states.

TFT Example

The following statement lines specify distributed defect states which would typically be used for polysilicon.

```
DEFECTS NTA=1.E21 NTD=1.E21 WTA=0.033 WTD=0.049 \
        NGA=1.5E15 NGD=1.5E15 EGA=0.62 EGD=0.78 \
        WGA=0.15 WGD=0.15 SIGTAE=1.E-17 \
        SIGTAH=1.E-15 SIGTDE=1.E-15 SIGTDH=1.E-17 \
        SIGGAE=2.E-16 SIGGAH=2.E-15 SIGGDE=2.E-15 \
        SIGGDH=2.E-16
```

DEGRADATION

DEGRADATION – specifies parameters for MOS device degradation modeling.

Syntax

```
DEGRADATION <params>
```

Parameter	Type	Default	Units
F.NTA	Char		
F.NTD	Char		
F.NTD	Char		
F.SIGMAE	Char		
F.SIGMAH	Char		
NTA	Real	$1.0x10^{11}$	cm^{-2}
NTD	Real	$1.0x10^{10}$	cm^{-2}
SIGMAE	Real	$1.0x10^{17}$	cm^{-2}
SIGMAH	Real	$1.0x10^{17}$	cm^{-2}

Description

NTA specifies the uniform acceptor-like trap density on the interface.

NTD specifies the uniform donor-like trap density on the interface.

SIGMAE specifies the acceptor-like trap capture cross section.

SIGMAH specifies the donor-like trap capture cross section.

F.NTA specifies the file name for a C-INTERPRETER function that specifies arbitrary density distribution of the acceptor-like traps on the interface.

F.NTD specifies the file name for a C-INTERPRETER function that specifies arbitrary density distribution of the donor-like traps on the interface.

F.SIGMAE specifies the name of a file containing a C-INTERPRETER function specifying the distribution of acceptor trap cross-sections.

F.SIGMAH specifies the name of a file containing a C-INTERPRETER function specifying the distribution of donor trap cross-sections.

MOS Interface State Example

```
DEGRADATION NTA=1.E-12 SIGMAE=5.E-18
```

This syntax defines a density of acceptor states uniformly distributed along the silicon-oxide interface. The trapping cross section is also defined. Traps will be filled by gate current in transient mode simulations leading to a shift in device parameters.

DOPING

DOPING — specifies doping profiles either analytically or from an input file.

Syntax

```
DOPING <prof>[<psp>][<bound>][<loc>][<sprea>>][OUTFILE= <fn>][<trps>]
```

Parameter	Type	Default	Units
ACTIVE	Logical	True	
ANTIMONY	Logical	False	
ARSENIC	Logical	False	
ASCII	Logical	False	
ATHENA	Logical	False	
ATHENA.1D	Logical	False	
BORON	Logical	False	
CHARACTERISTIC	Real		µm
CHEMICAL	Logical	False	
CONCENTRATION	Real	0	cm^{-3}
DEGEN	Real		
DEVICE	Char		1
DIRECTION	Character	y	
DOPOFF	Real	0	cm^{-3}
DOSE	Real		cm^{-2}
ERFC.LATERAL	Logical	False	
E.LEVEL	Real		eV
F.COMPOSIT	Character		
F.DOPING	Character		
F3.DOPING	Character		
F.OXCHARGE	Character		
GAUSSIAN	Logical	False	
INDIUM	Logical	False	
INFILE	Character		
JUNCTION	Real		µm
LAT.CHAR	Real		µm

Parameter	Type	Default	Units
MASTER	Logical	False	
N.TYPE	Logical	False	
NET	Logical	False	
OUTFILE	Character		
PHOSPORUS	Logical	False	
P.TYPE	Logical	False	
RATIO.LATERAL	Real	0.7	
REGION	Real		
RESISTI	Real		
OX.CHARGE	Logical	False	
REGION	Integer	All	
SIGN	Real		cm^2
SIGP	Real		cm^2
SLICE.LAT	Real		µm
START	Real	0	
SUPREM3	Logical	False	
TAUN	Real		
TAUP	Real		
TRAP	Logical	False	
UNIFORM	Logical	False	
X.COMP	Logical	False	
X.LEFT	Real	left of structure	µm
X.MAX	Real		µm
X.MIN	Real		µm
X.RIGHT	Real	right of structure	µm
Y.BOTTOM	Real	bottom of structure	µm
Y.COMP	Logical	False	
Y.MAX	Real		µm
Y.MIN	Real		µm
Y.TOP	Real	top of structure	
Z.BACK	Real		

Parameter	Type	Default	Units
Z.FRONT	Real		
Z.MIN	Real		
Z.MAX	Real		
ZLAT.CHAR	Real		
ZSLICE.LAT	Real		
ZRATIO.LAT	Real	0.7	

Description

The DOPING statement is used to define doping profiles in the device structure. Typically a sequence of DOPING statements is given each building on the others.

OUTFILE specifies the name of an output file for use with REGRID. The first DOPING statement should use this parameter to specify a filename. All doping information from the first DOPING statement and all subsequent DOPING statements in the input file are saved to this file. The REGRID statement can read this file and interpolate doping on the new grid.

Note: The file from OUTFILE cannot be used in TONPLOT or in the MESH statement.The SAVE command should be used after *all* of the DOPING commands required to save a file for plotting the doping profile.

Analytical Profile Types

These parameters specify how ATLAS will generate a doping profile from analytical functions.

GAUSSIAN specifies the use of a gaussian analytical function to generate the doping profile. If GAUSSIAN is specified, the following parameters must also be specified:

Polarity parameters N.TYPE or P.TYPE

One of the following groups of profile specifications:

Group 1:CONCENTRATION and JUNCTION

Group 2:DOSE and CHARACTERISTIC

Group 3:CONCENTRATION and CHARACTERISTIC

UNIFORM specifies the use of uniform (constant) analytical functions to generate the doping profile. If Uniform is specified, the N.TYPE, P.TYPE, and CONCENTRATION parameters must be specified. Doping is introduced into a box defined by the boundary parameters (see page 15-25). The box by default includes the entire region.

F.DOPING specifies the name of a file containing a C-INTERPRETER function specifying the spatial distribution of dopants.

F3.DOPING specifies the name of a file containing a C-INTERPRETER function specifying the spatial distribution of dopants for a 3D device.

File Import Profile Types

These parameters specify how **ATLAS** will generate a doping profile from a file. Files can be user-defined or from process simulation.

INFILE specifies the name of the appropriate input file.

ASCII has two separate meanings:

- when combined with other format parameters is specifies the filetype is ASCII
- when used alone, it specifies ASCII data files containing concentration versus depth information.

In the second meaning this parameter must be written in the form ASCII `INFILE=<filename>`, where `filename` is the name of the ASCII input file. The data file must be in the following format:

depth concentration

depth concentration

depth concentration

...

where **depth** is specified in μm and **concentration** is specified in cm^{-3}. An input file name, a dopant type, and boundary parameters must be specified. Positive concentrations are assumed to be n-type and negative concentrations are assumed to be p-type unless the `N.TYPE` or `P.TYPE` parameters are used.

MASTER specifies that the `INFILE` is written in the Silvaco standard structure file format. This file format is the default output format of ATHENA and SSUPREM3. This parameter is typically combined with the `SSUPREM3`, `ATHENA 1D` or `ATHENA` parameters. If neither of these are used the default is `SSUPREM3`.

SUPREM3 specifies the `INFILE` was produced by SSUPREM3 in standard structure file (SSF) format or binary or an ASCII export format. *Export format is an obsolete file format.* If this profile type is used, an input file name, a dopant, and boundary parameters must be specified. When SSUPREM3 produces an output file, the doping profiles are stored by dopant. Therefore, a dopant parameter should be specified in order to import the correct doping profile into ATLAS. If a specific dopant is not specified the total donors and acceptor concentrations are loaded.

ATHENA.1D specifies that the doping file is a ATHENA 1D export file. This parameter acts in a similar way to the `SSUPREM3` parameter.

ATHENA reads 2D doping information from ATHENA standard structure file (SSF) or PISCES-II format files. *The PISCES-II format is an obsolete file format.*. Doping information obtained from this file will be added to each point of the current ATLAS mesh. If points in the ATLAS mesh do not coincide with points in the ATHENA mesh, doping for ATLAS mesh points will be interpolated from ATHENA doping information. If this profile type is used, the `INFILE` parameter must also be specified.

Note: The `X.STRETCH` function available in previous versions of ATLAS has been replaced by similar more powerful functions in DEVEDIT. This feature should no longer be used in ATLAS.

Parameters that Specify the Dopant Type

These parameters give information about the dopant species or type to be used inthe specified profile. Different profile types require different profile specifications.

ACTIVE specifes that for the dopant specified the active concentration as opposed to the chemical concentration is added. This is true by default. Files from ATHENA or SSUPREM3 contain both active and chemical concentrations for each dopant.

ANTIMONY specifies that antimony dopant information be extracted from an imported file.

ARSENIC specifies that arsenic dopant information be extracted from an imported file.

BORON specifies that boron dopant information be extracted from an imported file.

CHEMICAL specifies that the chemical concentration (as opposed to the active concentration) will be read from the imported file. This is generally not advisable.

DOPOFF subtracts a background doping value from the ATHENA or SSUPREM3 doping.

INDIUM specifies that indium dopant information be extracted from an imported file.

NET specifies that net doping information be extracted from an imported file. This is usually not advisable. It is better to use several DOPING statements to extract data dopant by dopant from a file.

N.TYPE, DONOR specifies an n-type or donor dopant. This parameter may be used with GAUSSIAN and UNIFORM profile types.

P.TYPE, ACCEPTOR specifies a p-type or acceptor dopant. This parameter may be used with GAUSSIAN and UNIFORM profile types.

PHOSPHORUS specifies that phosphorus dopant information be extracted from an imported file.

TRAP specifies that the dopant concentration is to be treated as a trap state density.

OX.CHARGE specifies a fixed oxide charge profile. Oxide charge can only be placed in any insulator region. The N.TYPE/P.TYPE parameters are *not* used hence a negative concentration implies a negative charge.

X.COMP specifies a profile of composition fraction **x** as defined in Appendix B. This profile can be used to change the composition fraction of cations in terniary and quaterniary materials over a spatial distribution.

Y.COMP specifies a profile of composition fraction **y** as defined in Appendix B. This profile can be used to change the composition fraction of anions in terniary and quaterniary materials over a spatial distribution.

RESISTI can be used to specify resistivity, which is converted to carrier concentration in silicon (i.e. this parameter replaces the CONC parameter). This conversion uses tables of resistivity versus concentration for donors and acceptors. These tables currently make no distinction between specific changes of the same type. The Arora mobility model is the basis of these tables.

Vertical Distribution Parameters

CHARACTERISTIC specifies the principal characteristic length of the implant. If this parameter is left unspecified, the principal characteristic can be computed from the values of the

- Polarity Parameters
- Boundary Parameters
- Concentration and Junction parameters

CONCENTRATION specifies the peak concentration when a Gaussian profile is used. If this parameter is not specified, peak concentration may be computed from the values of the polarity, boundary, DOSE, or RESISTI, CHARACTERISTIC concentrations. When a uniform profile is specified, the CONCENTRATION parameter sets the value of the uniform doping level. Concentrations must be positive.

DOSE specifies the total dose for a Gaussian profile.

JUNCTION specifies the location of a p-n junction within the silicon region of a Gaussian profile. When JUNCTION is specified, the characteristic length is computed by examining the doping at a point halfway between the end of the constant box and the given depth. The JUNCTION location is evaluated considering all previous DOPING statements only. This means that in some cases the order of DOPING statements is important.

PEAK specifies the depth location of the peak doping in a Gaussian profile.

Location Parameters

DIRECTION specifies the axis along which a one-dimensional profile is directed in a two-dimensional device (x or y). DIR=y will typically be used for implanted profiles.

REGION specifies the region number where doping is to be added.

START specifies the depth in the y direction where the profile should start.

Lateral Extent Parameters

These parameters must be specified when a 1-D doping profile type is used (MASTER, GAUSSIAN, or ASCII). These boundary parameters set the doping boundaries before applying lateral spreading. This is equivalent to setting implant mask edges.

X.MIN, X.MAX, Y.MIN, Y.MAX, Z.MIN and **Z**.**MAX** specify the x, y and z bounds of a rectangular shaped region or box in the device. The dopant profile within this box will be constant with a density equal to the value specified by the CONC parameter. Outside this box the profile decreases from the peak, CONC, with distance, from the box along the principal axes. The relationship between the concentration, outside the box, to distance will depend upon the profile type as specified by the GAUSSIAN, MASTER, ATHENA, ATLAS, and UNIFORM parameters.

X.**LEFT, XMIN** specifies the left boundary of a vertical 1-D profile.

X.**RIGHT, X.MAX** specifies the right boundary of a vertical 1-D profile.

Y.**BOTTOM, Y.MAX** specifies the bottom boundary of a horizontal 1-D profile.

Y.**TOP, Y.MIN** specifies the top boundary of a horizontal 1-D profile.

Z.**BACK**, **Z.MIN** specifies the back boundary of a z directed 1-D or 2-D profile.

Z.**FRONT, Z.MAX** specifies the front boundary of a z directed 1-D or 2-D profile.

Lateral Distribution Parameters

These parameters specify how a vertical 1-D profile is extended outside the box defined by the lateral extent parameters.

LAT.**CHAR** specifies the characteristic length of the lateral profile. If this parameter is not specified, the characteristic length is defined by:

$$CL = RL \times OCL$$

where:

CL is the lateral characteristic length in the x direction.

RL is the value of RATIO.LATERAL.

OCL is the characteristic length of the original profile in the y direction.

ERFC.**LATERAL** specifies an error function used for lateral spreading. If two-dimensional spreading parameters are used in conjunction with a Gaussian profile, the lateral impurity profile may be transformed into an error function.

RATIO.**LATERAL** is the ratio of characteristic lengths in the **x** and **y** directions.

SLICE.**LAT** specifies the point at which the doping is examined to compute the characteristic length of a Gaussian profile after JUNCTION has been specified. The default for this parameter is a point halfway between the end of the constant box and the given depth.

ZLAT.**CHAR** specifies the characteristic length of the laterial profile in the **z** direction. See also LAT.CHAR.

ZRATIO.**LATERAL** is used analogously to RATIO.LATERAL but applies to lateral spreading in the **z** direction. See also LAT.CHAR.

ZSLICE.**LAT** is analogous to SLICE.LAT but applies to profiels in the **z** direction.

Trap Parameters

Note: See the TRAP statement for more information on each of these parameters

---

DEVICE specifies which device the statement applies to in MIXEDMODE simulation.

REGION specifies which region the traps apply to. If unspecified, the traps apply to all regions.

E.LEVEL sets the energy of the discrete trap level. For acceptors E.LEVEL is relative to the conduction band edge, for donors it is relative to the valence band edge.

DEGEN.FAC specifies the degeneracy factor of the trap level used to calculate the density.

SIGN speccifies the capture cross section of the trap for electrons.

SIGP specifies the capture cross section of the trap for holes.

TAUN specifies the lifetime of electrons in the trap level.

TAUP specifies the lifetime of holes in the trap level

Analytical Doping Defintion Example

This example describes a 1.0µm n-channel MOSFET using Gaussian source and drain profiles. The lateral extent of the source is given by X.RIGHT=2. This corresponds to the mask edge for the implant. Sub-diffusion is determined by an error function based on the RATIO.LAT and JUNCTION parameters. For both source and drain, the n+ doping is added to the uniform p-well concentration to ensure a junction depth of 0.3µm.

```
DOP UNIF CONC=1E16 P.TYPE
DOP GAUSS CONC=9E19 N.TYPE X.RIGHT=2 JUNC=0.3 RATIO.LAT=0.6 ERFC.LAT
DOP GAUSS CONC=9E19 N.TYPE X.LEFT=3  JUNC=0.3 RATIO.LAT=0.6 ERFC.LAT
```

1D ATHENA or SSUPREM3 Interface Example

This example reads a 1D ATHENA bipolar profile and adds it to a uniform substrate concentration. The base and emitter doping are loaded from the same file by specifying the impurity required for each area (boron in the base and arsenic in the emitter).

The DOPOFF parameter is used to subtract the substrate arsenic dopant out of the 1-D profile that is loaded since this dopant was already specified in the substrate doping line.

Versions of SSUPREM3 later than 5.0 use standard structure files as default when saving data. These can be loaded in ATLAS with the syntax below by replacing ATHENA.1D with SSUPREM3.

```
#   SUBSTRATE
DOP REGION=1 UNIF CONC=1E16 N.TYPE
#  BASE
DOP REGION=1 MASTER ATHENA.1D BORON RATIO.LAT=0.7 INF=bipolar.exp
 #  EMITTER
DOP REGION=1 MASTER ATHENA.1D ARSENIC RATIO.LAT=0.6 \
    INF=bipolar.exp X.LEFT=12.0 X.RIGHT=13.0 DOPOFF=1e16
```

Athena Doping Interface Example

This example demonstrates how to use an SSF format ATHENA file to interpolate doping onto a ATLAS grid and save the doping information for subsequent regrid operations.This is an alternative the prefered **Athena/Atlas** interface which is documented in the Getting Started Chapter.

```
DOPING ATHENA MASTER INFILE=NMOS.DOP OUTFILE=NMOS.DOP
REGRID DOPING ABS LOG RATIO=4 OUTFILE=NMESH1.STR DOPFILE=NMOS.DOP
```

3D Doping Defintion Example

The following example illustrates the formation of a Gaussian highly doped n-type area in a three-dimensional structure.

```
DOPING GAUSS N.TYPE CONC=1e20 PEAK=0.0 CHAR=0.2 X.LEFT=0.5 \
        X.RIGHT=1.0 Z.LEFT=0.5 Z.RIGHT=1.0
```

ELECTRODE

ELECTRODE — specifies the locations and names of electrodes in a previously defined mesh.

Syntax

```
ELECTRODE NAME=<en> [NUMBER=<n>] [SUBSTRATE] <pos> <reg>
```

Parameter	Type	Default	Units
BOTTOM	Logical	False	
IX.HIGH	Integer	right side of structure	
IX.LOW	Integer	left side of structure	
IY.HIGH	Integer	IY.LOW	
IY.LOW	Integer	top of structure	
IZ.HIGH	Real		
IZ.LOW	Real		
LEFT	Logical	False	
LENGTH	Real	length of structure	µm
MATERIAL	Character	Contact	
NAME	Character		
NUMBER	Integer	defined #(electrodes)+ 1	
RIGHT	Logical	False	
SUBSTRATE	Logical	False	
TOP	Logical	False	
X.MAX	Real	right side of structure	µm
X.MIN	Real	left side of structure	µm
Y.MAX	Real	Y.MIN	µm
Y.MIN	Real	top of the structure	µm
Z.MIN	Real		µm
Z.MAX	Real		µm

Description

MATERIAL specifies a material for the electrode (see Table B-1). This material will be displayed in TONYPLOT. The electrode material can also be used to define the electrode thermal characteristics (thermal conductivity) and optical characteristics (complex index of refraction). Setting the material here does not apply any electrical property such as workfunction to the terminal. All electrical properties of electrodes are set on the CONTACT statement.

NAME specifies an electrode name. Since other parts of the program use standard electrode names it is suggested that these names be used.The list of allowed names by other syntax is: **gate, gg, drain, dd, source, bulk, substrate, emitter, ee, collector, cc, base, bb, anode, cathode, fgate, cgate, ngate, pgate, well, nwell, pwell, channel,** and **ground**

NUMBER specifies an electrode number from 1 to 50. Electrode numbers may be specified in any order. If NUMBER is not specified, electrodes will be automatically numbered in sequential order. This parameter cannot renumber electrodes already defined in ATLAS or other programs.

SUBSTRATE places the specified electrode at the bottom of the device and names the electrode **substrate**.

pos is one of the position parameters described below.

reg is a set of the region parameters described on the next page.

Position Parameters

BOTTOM or **SUBSTRATE** specifies that the electrode is positioned along the bottom of the device.

LEFT specifies that the electrode starts at the left-hand edge of the device. The electrode will be positioned from left to right along the top of the device.

RIGHT specifies that the electrode starts at the right-hand edge of the device. The electrode will be positioned from right to left along the top of the device.

TOP specifies that the electrode is positioned along the top of the device.

Region Parameters

Device coordinates may be used to add regions to both rectangular and irregular meshes. In either case, boundaries must be specified with the X.MAX, X.MIN, Y.MAX, and Y.MIN parameters.

LENGTH specifies the length of the electrode in the x-direction. It is not necessary to specify X.MIN, X.MAX, and LENGTH. If two of these parameters are specified, the value of the third parameter willbe calculated.

X.**MAX** specifies the maximum x-boundary of the electrode.

X.**MIN** specifies the minimum x-boundary of the electrode.

Y.**MAX** specifies the maximum y-boundary of the electrode.

Y.**MIN** specifies the minimum y-boundary of the electrode..

Z.**MIN** specifies the minimum z-boundary of the electrode.

Z.**MAX** specifies the maximum z-boundary of the electrode.

Note: If an electrode has been shortened to fit the current mesh, a warning message will be generated by ATLAS. Electrode placement can only occur at previously defined mesh nodes.

Grid Indices

As an alternative to the region parameters above, it is permitted to use grid indices to define a region only when the mesh is rectangular although these parameters are not recommended. To define a region with a rectangular mesh:

1. The X.MESH and Y.MESH statements must be used to specify grid indices.

2. The IX.HIGH, IX.LOW, IY.HIGH, and IY.LOW parameters may be used to specify **x** and **y** values.

IX.**HIGH** specifies the maximum x-value of the grid index.

IX.**LOW** specifies the minimum x-value of the grid index.

IY.HIGH specifies the maximum y-value of the grid index

IY.LOW specifies the minimum y-value of the grid index.

Nodes which have **x** and **y** grid indices between IX.LOW and IX.HIGH and between IY.LOW and IY.HIGH are designated electrode nodes. Normally, horizontal planar electrodes will be used. In this case, IY.LOW equals IY.HIGH.

IZ.HIGH specifies the maximum z-value of the grid index.

IZ.LOW pecifies the minimum z-value of the grid index.

MOS Electrode Definition Example

This example defines electrodes for a typical MOS structure.

```
ELEC X.MIN=0.5 LENGTH=0.25 NAME=gate
ELEC LENGTH=0.25 Y.MIN=0 LEFT NAME=source
ELEC LENGTH=0.25 Y.MIN=0 RIGHT NAME=drain
ELEC SUBSTRATE
```

3D Electrode Definition Example

The following example illustrates electrode defintion for a 3-D structure.

```
ELECTRODE NAME=ANODE X.MIN=0.5 X.MAX=1.0 \
          Z.MIN=0.5 Z.MAX=1.0
```

Note: In ATLAS it is prefered to refer to ELECTRODES by name rather than number. However some functions might require the electrode number. The syntax MODELS PRINT can be used to echo electrode numbers to the run-time output.

ELIMINATE

ELIMINATE – terminates mesh points along lines in a rectangular grid defined within ATLAS in order to reduce the local mesh density.

Syntax

```
ELIMINATE X.DIRECTION|Y.DIRECTION [<boundary>]
```

Parameter	Type	Default	Units
COLUMNS	Logical	False	
IX.LOW	Integer		
IX.HIGH	Integer		
IY.LOW	Integer		
IY.HIGH	Integer		
ROWS	Logical	False	
X.MIN	Real		μm
X.MAX	Real		μm
Y.MIN	Real		μm
Y.MAX	Real		μm

Description

The ELIMINATE statement is used to remove points along every other line within the chosen range. Successive eliminations of the same range remove points along every fourth line. For horizontal elimination, the vertical bounds should be decreased by one at each re-elimination of the same region. For vertical elimination, the horizontal bounds should be decreased by one at each re-elimination of the same region.

ROWS or **X.DIR** eliminates points along horizontal lines.

COLUMNS or **Y.DIR** eliminates points along vertical lines.

Boundary Parameters

X.**MIN**, **X**.**MAX**, **Y**.**MIN**, and **Y**.**MAX** specify the location of the boundaries of an area in user coordinates where the elimination is applied.

The following are provided for backward compatibility only. Their use is not recommended.

IX.**HIGH** specifies the mesh line number high boundary in the x-direction.

IX.**LOW** specifies the mesh line number low boundary in the x-direction.

IY.**HIGH** specifies the mesh line number high boundary in the y-direction.

IY.**LOW** specifies the mesh line number low boundary in the y-direction.

Substrate Mesh Reduction Example

This example removes vertical points between the depth of 10μm and 20μm.

```
ELIM Y.DIR Y.MIN=10 Y.MAX=20 X.MIN=1 X.MAX=8
ELIM Y.DIR Y.MIN=10 Y.MAX=20 X.MIN=1 X.MAX=7
```

Note: In some cases, applications of the ELIMINATE statement can cause internal inconsistencies in the mesh. When this occurs, an error message will be issued warning of the existence of triangles that are not associated with any region.

Note: The ELIMINATE statement only works on meshes defined using ATLAS syntax. Elimination of mesh points on arbitrary meshes is possible in DEVEDIT

EXTRACT

EXTRACT statements – are used to measure parameters from both LOG and Solution files.

Note: These commands are executed by DECKBUILD. This statement is documented in the VWF INTERACTIVE TOOLS MANUAL VOLUME ONE.

Terminal Current Extraction Example

By default EXTRACT works on the currently open LOG file. For example to extract peak drain current from a run immediately after solution:

```
LOG OUTF=myfile.log
SOLVE .........
EXTRACT NAME="peak Id" max(i."drain")
```

Extraction Example from Previously Generated Results

To extract the same data from a previously run simulation the INIT parameter is required.

```
EXTRACT INIT INFILE="myfile.log"
EXTRACT NAME="peak Id" max(i."drain")
```

Solution Quantities Extraction Example

To use EXTRACT with solution files, it is **always** required to use the INIT parameter.To find the integrateed number of electrons in a 1D slice at X=1.0 use:

```
SAVE OUTF=mysolve.str   or   SOLVE ...... MASTER OUTF=mysolve.str
EXTRACT INIT INFILE="mysolve.str"
EXTRACT NAME="integrated e-" area from curve(depth,n.conc \
        material="Silicon" mat.occno=1 x.val=1.0)
```

Note: EXTRACT commands are generally case sensitive.

FOURIER

FOURIER — enables the ability to do Fourier transformations in ATLAS.

Syntax

```
FOURIER INFILE OUTFILE + [ OPTIONAL PARAMETERS ]
```

Parameter	Type	Default	Units
COMPLEX.VALUES	Logical	False	
FUNDAMENTAL	Real		Hz
INFILE	Character		
INTERPOLATE	Logical	False	
MAX.HARMONIC	Real		Hz
NUM.SAMPLES	Integer	64	
OUTFILE	Character		
T.START	Real		s
T.STOP	Real		s

Description

The FOURIER statement performs a Fast Fourier Transform on log file data.

COMPLEX.VALUES specifies that the real and imaginary components are saved to file as well as the magnitude and phase values (synonym: REAL.VALUES).

FUNDAMENTAL specifies the fundamental frequency. If this is not specified then the fundamental frequency is set to (T.STOP-T.START)/NUM.SAMPLES. If the fundamental frequency is specified then T.STOP is set to T.START + 1/FUNDAMENTAL.

INFILE specifies the input log file. This should contain data from a transient simulation.

INTERPOLATE specifies that the input data should be linearly interpolated such that data at uniform time steps are created. Interpolation of data can introduce addition (inaccurate) harmonic values into the Fourier transform. INTERPOLATE must be used if the log file contains non-uniform time steps.

MAX.HARMONIC specifies the maximum harmonic frequency that the Fourier transform should calculate. This will automatically calculate the correct number of samples (NUM.SAMPLES) required to generate this frequency. FUNDAMENTAL must be specified when MAX.HARMONIC is used.

NUM.SAMPLES specifies the number of discrete samples. This should be an integer power of 2, i.e. 2^n where n is a positive integer. The default value is 64 unless the MAX.HARMONIC parameter is specified. In this case the number of samples is set to the nearest integer power of 2 which will generate this frequency.

OUTFILE specifies the output file for the Fourier transform data.

T.START specifies the start of time data to be used for the Fourier transform. The default value is the first time point in the input log file.

T.STOP specifies the end of time data to be used for the Fourier transform. The default value is the last time point in the input log file.

Example 1:

In this example the transient data previously written to log file " hemt1.log " is transformed from the time domain to the frequency domain. The fundamental frequency is set to 0.5 GHz and harmonic frequencies up to 16 GHz are calculated. Since the data in " hemt1.log " has non-uniform time steps, the INTERPOLATE flag must be enabled. The complex values as well as the magnitude and phase values are stored in " fftout1.log ".

```
FOURIER INFILE=hemt1.log FUNDAMENTAL=5e8 MAX.HARMONIC=1.6E10 \
        OUTFILE=fftout1.log INTERPOLATE COMPLEX.VALUES
```

Example 2:

In this example the log file values between 31.25 ps and 2ns are transformed into the frequency domain. The fundamental frequency is automatically determined from the time period set by T.START and T.STOP. The data values from this time period are interpolated into 64 samples, giving a maximum harmonic frequency of 15.5 GHz. The magnitude and phase values are then stored in " fftout2.log ".

```
FOURIER INFILE=hemt1.log T.START=3.125e-11 T.STOP=2e-9 NUM.SAMPLES=64 \
        OUTFILE=fftout2.log INTERPOLATE
```

GO

GO - quits and restarts ATLAS and also defines certain global parameters for ATLAS execution

Note: This command is executed by DECKBUILD. This statement is documented in the VWF INTERACTIVE TOOLS MANUAL VOLUME ONE.

Example starting a given ATLAS Version

To start a given version of ATLAS the syntax is set by the simflags argument. To start version 4.3.0.R:

```
go atlas simflags="-V 4.3.0.R"
```

Parallel ATLAS Example

To define the number of processors to be used in parallel ATLAS the P flag is used. For example to start parallel ATLAS using 4 processors.

```
go atlas simflags="-V 4.3.2.C -P 4"
```

IMPACT

IMPACT — specifies and set parameters for impact ionization models.

Syntax

```
IMPACT <model>
```

Parameter	Type	Default	Units
A.NT	Real	0.588	
A.PT	Real	0.588	
AN1	Real	$7.03x10^{5}$	cm^{-1}
AN2	Real	$7.03x10^{5}$	cm^{-1}
AP1	Real	$6.71x10^{5}$	cm^{-1}
AP2	Real	$1.682x10^{6}$	cm^{-1}
B.NT	Real	0.248	
B.PT	Real	0.248	
BETAN	Real	1.0	
BETAP	Real	1.0	
BN1	Real	$1.231x10^{6}$	V/cm
BN2	Real	$1.231x10^{6}$	V/cm
BP1	Real	$1.231x10^{6}$	V/cm
BP2	Real	$2.036x10^{6}$	V/cm
C0	Real	$2.5x10^{-10}$	
CHIA	Real	$3.0x10^{5}$	
CHIB	Real	$5.0x10^{4}$	
CHI.HOLES	Real	$4.6x10^{4}$	
CROWELL	Logical	False	
CSUB.N	Real	$2.0x10^{14}$	
CSUB.P	Real	$4.0x10^{12}$	
DEVICE	Character		
EGRAN	Real	$4.0x10^{5}$	V/cm
ENERGY	Real	0.0025	
ETH.N	Real	1.8	eV

Parameter	Type	Default	Units
ETH.P	Real	3.5	eV
F.EDIIN	Character		
F.EDIIP	Character		
ICRIT	Real	$4.0x10^{-3}$	A/cm^2
INFINITY	Real	0.001	
LAMDAE	Real	$6.2x10^{-7}$	cm
LAMDAH	Real	$3.8x10^{-7}$	cm
LENGTH.REL	Logical	False	
LREL.EL	Real	$3.35x10^{-2}$	
LREL.HO	Real	$2.68x10^{-2}$	
M.ANT	Real	1.0	
M.APT	Real	1.0	
M.BNT	Real	1.0	
M.BPT	Real	1.0	
MATERIAL	Character		
NAME	Character		
REGION	Integer		
SELB	Logical	False	
TAUSN	Real	$0.4x10^{-12}$	s
TAUSP	Real	$0.4x10^{-12}$	s

Description

The impact ionization model for continuity equations allows the accurate prediction of avalanche breakdown for many devices. Since impact ionization is a two-carrier process, the following statement must be specified **after** setting impact ionization models:

```
METHOD CARRIERS=2
```

Model Selection Flags

CROWELL specifies the Crowell and Sze formulae[1].

SELB selects the impact ionization model described by Selberherr[3].

N.CONCANNON, P,CONCANNON set the Concannon substrate current model

Note: If no model selection flag is set then the model parameters from Grant are used. See the Physics Chapter for details.

Model Localization Parameters

DEVICE specifies the device in MIXEDMODE simulation to which the statement should apply.

MATERIAL specifies what material from the Table B-1 the statement should apply. If a material is specified then all regions defined as being composed of that material will be affected.

NAME specifies what region that the IMPACT statement should apply. Note that the name must match the name specified in the NAME parameter of the REGION statement.

REGION specifies that index of the region to which the impact parameters apply.

Crowell Model Parameters

LAMBDAE specifies the mean free path for electrons

LAMDAH specifies the mean free path for holes .

Selberrherr Model Parameters

AN1, AN2, BN1, BN2, EGRAN specify the basic set of parameters for Selberherr's impact ionization model. Index 1 (AP1, BP1, AN1, and BN1) corresponds to field values less than EGRAN, and index 2 (AP2, BP2, AN2, and BN2) corresponds to field values greater than EGRAN.

BETAN for electrons and **BETAP** for holes correspond to coefficients for the power of ECRIT/E

Temperature Dependence Parameters

A.NT specifies the value of the temperature-dependent parameter in Equation 3-240**.**

A.**PT** specifies the value of the temperature-dependent parameter in Equation 3-241.

B.**NT** specifies the value of the temperature-dependent parameter in Equation 3-242.

B.**PT** specifies the value of the temperature-dependent parameter in Equation 3-243.

M.**ANT** specifies the value of the temperature-dependent parameter in Equation 3-240.

M.**APT** specifies the value of the temperature-dependent parameter in Equation 3-941

M.**BNT** specifies the value of the temperature-dependent parameter in Equation 3-242.

M.**BPT** specifies the value of the temperature-dependent parameter in Equation 3-243.

Parameters for use with Energy Balance

Note: When energy balance simulations are run the Toyabe impact ionization model is used. This model is used irrespective of the settings SELB or CROWELL. See the Physics Chapter for details

F.**EDIIN** specifies the name of the file containing a C-INTERPRETER function describing the values of the parmeters in Equation 3-238 as a function of electron temperature.

F.**EDIIP** specifies the name of the file containing a C-INTERPRETER function describing the values of the parameters in Equation 3-239 as a function of hole temperature.

LENGTH.**REL** specifies the use of energy relaxation length for the impact ionization model with the energy balance model.

LREL.EL specifies an energy relaxation length for electrons if LENGTH.REL is specified.

LREL.HO specifies an energy relaxation length for holes if LENGTH.REL is specified.

TAUSN specifies the relaxation time for electrons in the temperature dependent impact model.

TAUSP specifies the relaxation time for holes in the temperature dependent impac model.

Concannon Model Parameters

CSUB.N is an empirical tuning factor used in Concannon's Substrate Current Model (eq. 3-264) for electrons

CSUB.P is an empirical tuning factor used in Concannon's Substrate Current Model (eq. 3-265) for holes

ETH.N specifies the ionization threshold energy for electrons used in Concannon's Substrate Current Model (eq.3-264)

ETH.P specifies the ionization threshold energy for holes used in Concannon's Substrate Current Model (eq.3-265)

C0 specifies the electron distribution weight factor used in Concannon's Substrate Current Model (eq.3-268)

CHIA specifies the electron distribution function constant used in Concannon's Substrate Current Model (eq. 3-268)

CHIB specifies the electron distribution function constant used in Concannon's Substrate Current Model (eq. 3-268)

CHI.HOLES specifies the hole distribution function constant used in Concannon's Substrate Current Model (eq. 3-269)

ENERGY.STEP specifies the energy step for numeric integration used in Concannon's Substrate Current Model

INFINITY specifies the limit for the highest energy in numeric integration used in Concannon's Substrate Current Model

Selberrherr Model Example

This example shows an IMPACT statement which specifies all parameters used by the model selected by the SELB parameter. In this case, only parameters for holes are field dependent. Parameters AP1 and BP1 correspond to parameters at field values more than EGRAN. Parameters AP2 and BP2 correspond to field values less than EGRAN. Coefficients for electrons should be repeated.

```
IMPACT SELB AN1=7.03E5 AN2=7.03E5 BN1=1.231E6 \
       BN2=1.231E6 AP1=6.71E5 AP2=1.58E6 BP1=1.693E6 \
       BP2=2.036E6 BETAN=1 BETAP=1 EGRAN=4.0E5
```

INTERFACE

INTERFACE — specifies interface parameters at semiconductor/insulator boundaries. All parameters apply only at the boundary nodes except where stated.

Syntax

```
INTERFACE [<params>]
```

Parameter	Type	Default	Units
AR.INDEX	Real	1	none
AR.THICK	Real	0	µm
CHARGE	Real	0.0	cm^{-2}
DEVICE	Char		cm/s
F.QF	Char		
S.N	Real	0.0	cm/s
S.P	Real	0.0	cm/s
QF	Real	0.0	cm^{-2}
REGION	Integer		
THERMIONIC	Logical	False	
X.MAX	Real	right hand side of structure	µm
X.MIN	Real	left hand side of structure	µm
Y.MAX	Real	bottom of structure	µm
Y.MIN	Real	top of structure	µm
Z.MIN	Real		µm
Z.MAX	Real		µm

Description

The INTERFACE statement consists of a set of boundary condition parameters for the interface and a set of parameter to localise the effect of these parameters.

Boundary Condition Parameters

S.**N** specifies the electron surface recombination velocity.

S.**P** specifies the hole surface recombination velocity.

QF specifies the fixed charge density.

CHARGE specifies interface charge density.

F.**QF** specifies the name of a file containing a C-INTERPRETER function describing the density of the interface fixed charge as a function of position.

THERMIONIC specifies that carrier transport across the interface by determined by thermionic emission. See the BLAZE Chapter for details. This parameter applied only at semiconductor/semiconductor boundaries.

AR.INDEX specifies the real component refractive index for the anti-reflective coating model in LUMINOUS. This model assumes the imaginary component of refractive index is zero so that the layer is non-absorbing. This model is only applicable for normal illumination.

AR.THICK specifies the thickness of an anti-reflective coating layer for the reflection model in LUMINOUS. This layer should generally not exist in the device mesh structure.

Position Parameters

X.**MIN**, **X**.**MAX**, **Y**.**MIN**, and **Y**.**MAX** define a bounding box. Any semiconductor/insulator interfaces found within this region are charged. If there is only one interface in a device, a nonplanar surface may be defined using a box which contains the whole device

X.**MIN** specifies the left x-coordinate of the bounding box.

X.**MAX** specifies the right x-coordinate of the bounding box.

Y.**MIN** specifies the bottom y-coordinate of the bounding box.

Y.**MAX** specifies the top y-coordinate of the bounding box.

Z.**MIN** specifies the front z-coordinate of the bounding box. It is used in 3-D modules only.

Z.**MAX** specifies the back z-coordinate of the bounding box. It is used in 3-D modules only.

DEVICE specifies which device in a MIXEDMODE simulation the statement applies to

REGION specifies which the region number the statement applies to.

MOS Example

This example defines an interface with both fixed charge and recombination velocities.

```
INTERFACE X.MIN=-4  X.MAX=4 Y.MIN=-0.5 Y.MAX=4 \
          QF=1E10 S.N=1E4 S.P=1E4
```

SOI Example

To define different fixed charge on the front and back interfaces of an SOI transistor two INTERFACE statements are required. In the syntax below the first statement will apply 5.10^{10} cm^{-2} charge to any silicon/oxide interface above Y=0.01µm. The second statement applied a higher charge to any interface below Y=0.01µm. Note that charges are only applied at the material interfaces so the Y-coordinate needs only to be somewhere within the silicon film.

```
INTERFACE Y.MAX-0.01 QF=5e10
INTERFACE Y.MIN=0.01 QF=2e11
```

Interface Charge for III-V Devices

Interface charge can be at the interfaces between two semiconductor regions or at the edges of semiconductor regions using the CHARGE parameter. The location of the added charge can be controlled using the **Position Parameters** described above.

INTTRAP

INTTRAP — activates interface defect traps at discrete energy levels within the bandgap of the semiconductor and sets their parameter values.

Syntax

```
INTTRAP <type> E.LEVEL=<r> DENSITY=<r> <capture parameters>
```

Parameter	Type	Default	Units
ACCEPTOR	Logical	False	
DEGEN.FAC	Real	1	
DENSITY	Real		cm^{-2}
DEVICE	Char		
DONOR	Logical	False	
E.LEVEL	Real		eV
F.DENSITY	Char		
REGION	Integer		
SIGN	Real		cm2
SIGP	Real		cm2
TAUN	Real		s
TAUP	Real		s

Description

DEVICE specifies which device the statement applies to in MIXEDMODE simulation.

DONOR specifies a donor-type trap level.

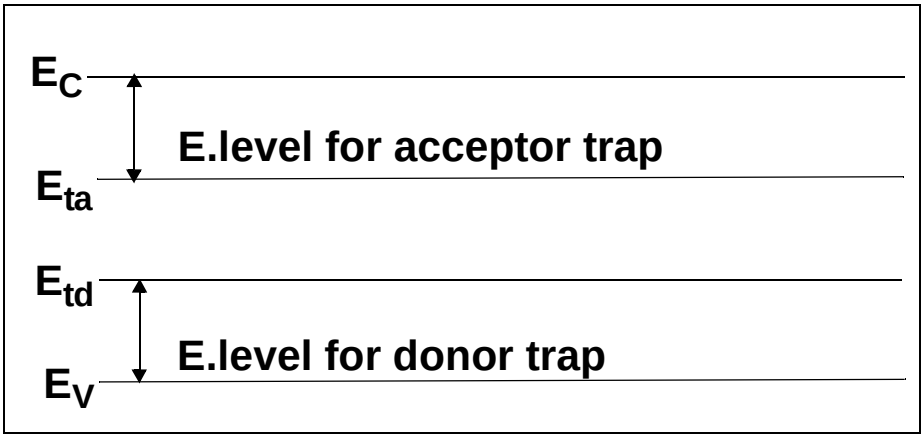
ACCEPTOR specifies an acceptor-type trap level.

DEGEN.FAC specifies the degeneracy factor of the trap level used to calculate the density.

DENSITY sets the maximum density of states of the trap level.

F.DENSITY specifies the name of a file containing a C-INTERPRETER function describing the density of donor/acceptor interface traps as a function of position.

E.LEVEL sets the energy of the discrete trap level. It is equal to the energy distance between conductance band and trap level for acceptor trap, and to energy distance between trap level and valence band for donor trap.



Capture Parameters

Either the cross section or lifetime parameters should be used to define the capture parameters.

SIGN speccifies the capture cross section of the trap for electrons.

SIGP specifies the capture cross section of the trap for holes.

TAUN specifies the lifetime of electrons in the trap level.

TAUP specifies the lifetime of holes in the trap level.

Example setting Multiple Interface Trap States

The following example sets three discrete interface trap levels within the silicon bandgap. These trap levels will capture carriers, slowing the switching speed of any device. In this example the capture cross sections are used to define the properties of each trap.

```
inttrap e.level=0.49 acceptor density=2.e10 degen=12 \
      sign=2.84e-15 sigp=2.84e-14
inttrap e.level=0.41 acceptor density=1.e10 degen=12 \
      sign=7.24e-16 sigp=7.24e-15
inttrap e.level=0.32 donor density=1.e10 degen=1 \
      sign=1.00e-16 sigp=1.00e-17
```

Note: For semiconductor bulk trap levels, see the TRAP statement.

LOAD

LOAD — loads previous solutions from files as initial guesses to other bias points.

Syntax

```
LOAD [ASCII|MASTER] [NO.CHECK] <files>
```

Parameter	Type	Default	Units
ASCII	Logical	False	
INFILE	Character		
INIFILE	Character		
IN2FILE	Character		
MASTER	Logical	False	
NO.CHECK	Logical	False	
TWOD	Logical	False	

Description

ASCII specifies that any original PISCES format files read or written by this statement will be in an ASCII rather than in a binary format.

MASTER specifies that any files read by this statement will be in a standard structure file rather than the original PISCES format. If you are using TONYPLOT to plot simulation results, this parameter should be specified.

TWOD allows loading of a 2-D solution into a 3-D structure. Note : The values from the 2-D solution are loaded uniformly in the Z direction.

NO.CHECK prevents checking material parameter differences between loaded binary files and the values set in the current input file.

File Parameters

The LOAD statement requires that one of the following file parameter syntaxes be used.

```
LOAD INFILE=<filename>

        or

LOAD IN1FILE=<filename> IN2FILE=<filename>
```

INFILE specifies a single input filename for solution data. This parameter should be used when you wish to load only one solution which is the most common case.

IN1FILE specifies a filename for present solution data. Use this parameter if two input files are needed to perform an extrapolation for an initial approximation (i.e., the PROJECT parameter of the SOLVE statement).

IN2FILE specifies an input filename for previous solution data. Use this parameter if two input files are needed to perform an extrapolation for an initial approximation (i.e., the PROJECT parameter of the SOLVE statement). The solution specified by this parameter is the first to be overwritten when new solutions are obtained.

Simple Save and Load Examples

This example saves and later loads master format solution file SOL.STR

```
SAVE OUTF=SOL.STR.
....
LOAD INFILE=SOL.STR MASTER
```

As before but using the SOLVE syntax

```
SOLVE OUTF=SOL.STR MASTER
..
LOAD INF=SOL.STR MASTER
```

When the save and load operations are not done within the same ATLAS run see the **Note** below.

Binary Format Example

Saving and loading using the binary format. This is quicker but these files cannot be plotted in TONYPLOT.

```
SOLVE OUTF=SOLVE_TMP
 ..
LOAD INF=SOLVE_TMP
```

Note: The function to calculate the difference between two files is now inside TONYPLOT. It has been discontinued from the LOAD statement

---

Note: The LOAD statement loads only the saved solution quantities into ATLAS. The mesh, electrodes, doping, regions, contact settings, material parameters, models and numerical methods must all be specified in advance of any LOAD statement. See the Re-Initializing ATLAS section of the Getting Started Chapter.

LOG

LOG — allows all terminal characteristics of a run to be lsaved to a file. Any DC, transient or AC data generated by SOLVE statements after the LOG statement is saved. Any parameters specified by the PROBE statement are also stored in the logfile. If a log file is already open, the open log file is closed and a new log file is opened.

Syntax

```
LOG [OUTFILE=<filename>] [MASTER] [acparams]
```

Parameter	Type	Default	Units
ABCD.PARAM	Logical	False	
APPEND	Logical	False	
GAINS	Logical	False	
H.PARAM	Logical	False	
IMPEDANCE	Real	50	ohms
INPORT	Character		
IN2PORT	Character		
J.ELECTRON	Logical	False	
J.HOLE	Logical	False	
LCOMMON	Real	0	H
LGROUND	Real	0	H
LIN	Real	0	H
LOUT	Real	0	H
MASTER	Logical	True	
OFF	Logical	False	
OLD	Logical	False	
OUTPORT	Character		
OUT2PORT	Character		
OUTFILE	Character		
RCOMMON	Real	0	ohms
RGROUND	Real	0	ohms
RIN	Real	0	ohms
ROUT	Real	0	ohms
S.PARAM	Logical	False	
SIM.TIME	Logical	False	s

Parameter	Type	Default	Units
WIDTH	Real	1	microns
Y.PARAM	Logical	False	
Z.PARAM	Logical	False	

File Output Parameters

OUTFILE specifies the log file that will be used to store DC, AC or transient I-V information.

MASTER specifies that AC data and I-V information will be saved in a standard structure file format. This is the default format.

J.ELECTRON specifies that electron currents are to be written into the log file.

J.HOLE specifies that hole currents are to be written into the log file.

APPEND specifies that the output I-V information should be appended to an existing log file. Care should be taken to ensure that the existing log files contain the same type of data (eg. DC, AC, transient) as the subsequent SOLVE statements.

OFF specifies that any currently open log file will be closed and log file output is discontinued.

OLD specifies that AC data and IV information will be saved in the original PISCES-II file format.

Synonym: PISCES

Note: The older ACFILE syntax is not supported and should not be used. AC results are stored in the file specified by OUTFILE as long as the first SOLVE statement after the LOG statement contains AC analysis.

RF Analysis Parameters

If S.PARAM, H.PARAM, Z.PARAM, GAINS or ABCD.PARAM is specified, the capacitance and conductance data will be converted into the requested set of AC parameters.

S.PARAM elects s parameter analysis. For s-parameter analysis the user may also choose to set any of the parasitic element parameters.

H.PARAM selects h parameter analysis.

Y.PARAM selects Y parameter analysis

Z.PARAM selects z parameter analysis.

ABCD.PARAM selects ABCD parameter analysis.

GAINS selects the calculation of several types of gains used in RF analysis[139]. These are the stability factor, unilateral power gain (GUmax), maximum unilateral transducer power gain (GTmax), maximum available power gain (Gma) and the maximum stable power gain (Gms). The magnitude of H21 is also calculated.

IMPEDANCE specifies the matching impedance for s-parameter calculation.

INPORT specifies the electrode name for the primary input port used when performing any AC parameter calculations.

IN2PORT specifies the electrode name of the secondary input port.

OUTPORT specifies the electrode name for the output ports used when performing any AC -parameter calculations.

OUT2PORT specifies the electrode n of the secondary output port

Parasitic Element Parameters

For RF parameter extraction, the user may also set any of the parasitic element parameters. By setting the parasitic element parameters, the user can apply lumped parasitic resistances or inductances to the terminal of the two-port device during the RF parameter extraction. These parameters will not affect the capacitance or conductance matrices calculated by ATLAS.

RIN specifies the lumped parasitic resistance on the input to the two port device for s-parameter extraction. The value of RIN is in Ohms.

ROUT specifies the lumped parasitic resistance on the output to the two port device for s-parameter extraction. The value of ROUT is in Ohms.

RGROUND specifies the lumped parasitic resistance on the ground or common side of the two port device for s-parameter extraction. The value of RGROUND is in Ohms. RCOMMON is an alias for RGROUND.

LIN specifies the lumped parasitic inductance on the input to the two port device for s-parameter extraction. The value of LIN is in Henrys.

LOUT specifies the lumped parasitic inductance on the output to the two port device for s-parameter extraction. The value of LOUT is in Henrys.

LGROUND specifies the lumped parasitic inductance on the ground or common side of the two port device for s-parameter extraction. The value of LGROUND is in Henris. LCOMMON is an alias for LGROUND.

SIM.TIME saves the time taken for a bias point into the log file and measured in seconds. Note that if multiple jobs are using the same cpu that this method may not be a true reflection of processor speed.

WIDTH specified an output width (in z-direction) to apply during the s-parameter calculation. Note that this parameter affects *only* the derived RF parameters and not currents, capacitances or conductances. The WIDTH parameter of the MESH statement can be used to scale these. Using both these WIDTH parameters will lead to a multiplication of the two widths for the RF parameters.

Simple Logfile Defintion Example

This example saves all I-V data in file myfile.log

```
LOG OUTF=myfile.log
```

Results should be plotted using TONYPLOT.

RF Analysis Example

To generate s-parameters assuming the input is gate/source and the output is drain/source. A width of 100 microns is also defined along with 100ohm resistance on the input:

```
LOG OUTF=mysparams.log S.PARAM INPORT=gate OUTPORT=drain \
    IN2PORT=source OUT2PORT=source WIDTH=100 RIN=100
```

Transient or AC Logfile Example

The contents of LOG files varies for different types of simulations (eg. DC, transient, AC). The content is set by the first SOLVE statement after the LOG statement. Therefore the following syntax is required

```
SOLVE VDRAIN=0.5

LOG OUTF=myfile.log

SOLVE VDRAIN=3.0 RAMPTIME=1e-9 DT=1e-11 TSTOP=1e-7
```

Correct transient parameters would not have been stored if the LOG statement had been placed before the first SOLVE statement which is DC.

LX.MESH, LY.MESH

L<n>.MESH — specifies the location of grid lines along the <n>-axis in a rectangular mesh used in LASER simulation. The syntax is equivalent for x and y directions.

Syntax

```
LX.MESH NODE=<n> LOCATION=<n>
```

Parameter	Type	Default	Units
LOCATION	Real		μm
NODE	Integer		

Description

NODE specifies the mesh line index. These mesh lines are assigned consecutively.

LOCATION specifies the location of the grid line.

LASER Mesh Example

This syntax defines a mesh of 33x33 covering the area bounded by (0.3,0.0) to (2.4,1.0)

```
LX.M n=1 l=0.3
LX.M n=33 l=2.4
LY.M n=1 l=0.0
LY.M n=33 l=1.0
```

Note: The mesh defined in these statements for the LASER Helmholtz Solver is entirely separate from the electrical device simulation mesh defined on the MESH statement.

MATERIAL

MATERIAL — associates physical parameters with materials in the mesh. The parameter default values for standard semiconductors are shown in Appendix B.

Syntax

```
MATERIAL <localization>  <material_definition>
```

Parameter	Type	Default	Units
AFFINITY	Real	see Appendix B	eV
ALIGN	Real	use AFFINITY	
ALPHAA	Real	0.0	
ALPHAR	Real	4.0	
ARICHN	Real	see Appendix B	$A/cm^2/K^2$
ARICHP	Real	see Appendix B	$A/cm^2/K^2$
ASYMMETRY	Real	0.5	
AUGN	Real	see Appendix B	cm^6/s
AUGP	Real	see Appendix B	cm^6/s
AUGKN	Num	0.0	cm^3
AUGKP	Num	0.0	cm^3
BGN.C	Real	0.5	
BGN.N	Real	$1.0x10^{17}$	cm^{-3}
BGN.E	Real	$9.0x10^{-3}$	V
COPT	Real	0.0	cm^3/s
DEVICE	Character		
EAB	Real	0.045	eV
EDB	Real	0.044	eV
EG300	Real	see Appendix B	eV
EGALPHA	Real	see Appendix B	eV
EGBETA	Real	see Appendix B	K
ETRAP	Real	0.0	eV
EPSINF	Real		eV/K
F.ALPHAA	Character		
F.TAUN	Character		

Parameter	Type	Default	Units
F.TAUP	Character		
F.TAURN	Character		
F.TAURP	Character		
F.MUNSAT	Character		
F.MUPSAT	Character		
F.BANDCOMP	Character		
F.EPSILON	Character		
F.VSATN	Character		
F.VSATP	Character		
F.RECOMB	Character		
F.INDEX	Character		
F.BGN	Character		
F.CONMUN	Character		
F.CONMUP	Character		
F.COPT	Character		
F.GAUN	Character		
F.GAUP	Character		
FCN	Real	$3.0x10^{-18}$	cm^{-2}
FCP	Real	$7.0x10^{-18}$	cm^{-2}
FERRO.EC	Real	0.0	V/cm
FERRO.EPSF	Real	1.0	
FERRO.PS	Real	0.0	C/sqcm
FERRO.PR	Real	0.0	C/sqcm
GAIN0	Real	2000.0	cm^{-1}
GAIN00	Real	-200.0	cm^{-1}
GAININ	Real	0	cm^2
GAINIP	Real	0	cm^2
GAIN1MIN	Real	$3.0x10^{-16}$	cm^2
GAIN2NP	Real	0	cm^5
GAMMA	Real		
GCB	Real	2.0	

Parameter	Type	Default	Units
G.SURF	Real	1.0	cm^2
GVB	Real	4.0	
G.CONDUCTIVITY	Real		
HC.A	Real	See Appendix B	J/Kcm^3
HC.B	Real	See Appendix B	J/K^2cm^3
HC.C	Real	See Appendix B	J/Kcm^3
HC.D	Real	See Appendix B	$JKcm^3$
IMAG.INDEX	Real	See Appendix B	
INDEX.FILE	Character		
KLASRH	Logical	False	
KLAAUG	Logical	False	
KSRHTN	Real	$2.5x10^{-3}$	s
KSRHTP	Real	$2.5x10^{-3}$	s
KSRHCN	Real	$3.0x10^{-13}$	cm^3
KSRHCP	Real	$11.76x10^{-13}$	cm^3
KSRHGN	Real	1.77	
KSRHGP	Real	0.57	
KAUGCN	Real	$1.83x10^{-31}$	cm^6/s
KAUGCP	Real	$2.78x10^{-31}$	cm^6/s
KAUGDN	Real	1.18	
KAUGDP	Real	0.72	
LT.TAUN	Real	0.0	
LT.TAUP	Real	0.0	
MASS.VTHN	Real	0.0	
MASS.VTHP	Real	0.0	
MATERIAL	Character		
MUN	Real	See Appendix B	
MUP	Real	See Appendix B	
NAME	Character		
NC300	Real	See Appendix B	cm^{-3}

Parameter	Type	Default	Units
NSRHN	Real	See Appendix B	cm^{-3}
NSRHP	Real	See Appendix B	cm^{-3}
NUE.EXTR	Real		
NUH.EXTR	Real		
NV300	Real	See Appendix B	cm^{-3}
OXCH.ONLY	Logical	False	
PERMITTIVITY	Real	See Appendix B	F/cm
POWER	Real		
REAL.INDEX	Real	See Appendix B	
REGION		All regions	
SEMICONDUC	Logical	False	
TAUMOB.EL	Real	$0.4x10^{-12}$	s
TAUMOB.HO	Real	$0.4x10^{-12}$	s
TAUN0	Real	See Appendix B	s
TAUP0	Real	See Appendix B	s
TAUREL.EL	Real	$0.4x10^{-12}$	s
TAUREL.HO	Real	$0.4x10^{-12}$	s
TC.A	Real	See Appendix B	(cm·K)/W
TC.B	Real	See Appendix B	cm/WK
TC.C	Real	See Appendix B	cm/WK
TC.C0	Real		
TC.NPOW	Real		
TC.E	Real		
TC.D	Real		
TCON.CONST	Logical		
TCON.POWER	Logical		
TCON.POLYNOM	Logical		
TCON.RECIPRO	Logical		
TMUN	Real		
TMUP	Real		
TRE.T1	Real		s

Parameter	Type	Default	Units
TRE.T2	Real		s
TRE.T3	Real		s
TRE.W1	Real		eV
TRE.W2	Real		eV
TRE.W3	Real		eV
TRH.T1	Real		s
TRH.T2	Real		s
TRH.T3	Real		s
TRH.W1	Real		eV
TRH.W2	Real		eV
TRH.W3	Real		eV
VSAT	Real		cm/s
VSATN	Real		cm/s
VSATP	Real		cm/s
VSATURATION	Real		cm/s
VTHN	Real	See Appendix B	cm/s
VTHP	Real	See Appendix B	cm/s

Description

The MATERIAL statment is used set basic material parameters related to band structure and parameters for certain mobility, recombination or carrier statistics models. Parameters for temperature dependence are noted in a separate section below.

Localization of Material Parameters

DEVICE specifies which device the MATERIAL statement should apply to in MIXEDMODE simulation.

MATERIAL specifies which material from the table in Appendix B that the MATERIAL statement should apply. If a material is specified, then all regions defined as being composed of that material will be affected.

NAME specifies which region the MATERIAL statement should apply. Note that the name must match the name specified in the NAME parameter of the REGION statement.

REGION specifies the region number to which these parameters apply. If there is more than one semiconductor region, specification of different parameters for each region is allowed. If REGION is not specified, all regions in the structure are changed.

Band Structure Parameters

AFFINITY specifies the electron affinity.

ALIGN specifies the fraction of the bandgap difference that is applied to the conduction band edge, relative to the minimum bandgap material in the device. Note that specifying this parameter overrides any electron affinity specification. See the BLAZE chapter for notes on setting the band alignment.

ARICHN specifies the effective Richardson constant for electrons.

ARICHP specifies the effective Richardson constant for holes.

EG300 specifies energy gap at 300K (see Equation 3-36).All semiconductor materials in ATLAS must have a defined EG300.

F.BANDCOMP specifies the name of a file containing a C-INTERPRETER funciton for the specification of temperature and composition dependent band parameter models.

F.EPSILON specifies the name of a file containing a C-INTERPRETER function for the specification of temperature and composition dependent static dielectric constant models.

MASS.VTHN specifies the electron effective mass for calculation of thermal velocity in the thermionic heterojunction model (see Equation 5-45).

MASS.VTHP specifies the hole effective mass for calculation of thermal velocity in the thermionic heterojunction model (see Equation 5-46).

NC300 specifies the conduction band density at 300K. (see Equation 3-29).

NV300 specifies valence band density at 300K (see Equation 3-30).

PERMITTIVITY specifies dielectric permittivity of the material. All materials in an ATLAS structure must have a defined permittivity.

Mobility Model Parameters

F.CONMUN specifies the name of a file containing a C-INTERPRETER function for the specification of temperature, composition and doping dependent electron mobility models.

F.CONMUP specifies the name of a file containing a C-INTERPRETER function for the specification of temperature, composition and doping dependent hole mobility models.

F.MUNSAT specifies the name of a file containing a C-INTERPRETER function for the specification of parallel field dependent electron mobility model for velocity saturation.

F.MUPSAT specifies the name of a file containing a C-INTERPRETER function for the specification of parallel field dependent hole mobility model for velocity saturation.

F.VSATN specifies the name of a file containing a C-INTERPRETER function for the specification of temperature and compostion dependent electron saturation velocity models.

F.VSATP specifies the name of a file containing a C-INTERPRETER function for the specification of temperature and compostion dependent hole saturation velocity models.

GSURF specifies a factor by which mobility is reduced at the semiconductor surface.This is a simple but not accurate alternative to the transverse field dependent or surface mobility models set on the MODELS statement.

MUN specifies low-field electron mobility. This parameter is only used if no concentration dependent mobility model is specified.

MUP specifies low-field hole mobility.This parameter is only used if no concentration dependent mobility model is specified.

VSATURATION specifies the saturation velocity for the electric field dependent mobility.

VSATN specifies the saturation velocity for electrons.

VSATP specifies the satruation velocity for holes.

Recombination Model Parameters

AUGN specifies the Auger coefficient cn (see Equation 3-227).

AUGP Specifies the Auger coefficient cp (see Equation 3-227).

AUGKN parameter of the narrow band-gap electron Auger recombination coefficient model.

AUGKP parameter of the narrow band-gap electron Auger recombination coefficient model.

COPT specifies the optical recombination rate for the material. This parameter has no meaning unless MODELS OPTR has been specified (see Equation 3-226).

ETRAP specifies the trap energy for SRH recombination

F.COPT specifies the name of a file containing a C-INTERPRETER function for the specification of composition and temperature dependence of the radiative recombination rate.

F.GAUN specifies the name of a file containing a C-INTERPRETER function for the specification of composition and temperature dependence of the electron Auger coefficient.

F.GAUP specifies the name of a file containing a C-INTERPRETER function for the specification of composition and temperature dependence of the hole Auger coefficient.

F.TAUN specifies the name of a file containing a C-INTERPRETER function for the specification of position dependent electron lifetime models.

F.TAUP specifies the name of a file containing a C-INTERPRETER function for the specification of position dependent hole lifetime models.

F.TAURN specifies the name of a file containing a C-INTERPRETER function specifying the electron relaxation time as a function of electron energy.

F.TAURP specifies the name of a file containing a C-INTERPRETER function specifying the hole relaxation time as a function of hole energy.

F.RECOMB specifies the name of a file containing a C-INTERPRETER function for the specification of temperature, composition, electron and hole concentration dependent recombination rate models.

KLASRH and KLAAUG Model parameters

KSRHTN coefficient for Klaassen's concentration and temperature dependent SRH lifetime model.

KSRHTP coefficient for Klaassen's concentration and temperature dependent SRH lifetime model.

KSRHCN coefficient for Klaassen's concentration and temperature dependent SRH lifetime model.

KSRHCP coefficient for Klaassen's concentration and temperature dependent SRH lifetime model.

KSRHGN coefficient for Klaassen's concentration and temperature dependent SRH lifetime model.

KSRHGP coefficient for Klaassen's concentration and temperature dependent SRH lifetime model.

KAUGCN coefficient for Klaassen's concentration dependent Auger model.

KAUGCP coefficient for Klaassen's concentration dependent Auger model.

KAUGDN coefficient for Klaassen's concentration dependent Auger model.

KAUGDP coefficient for Klaassen's concentration dependent Auger model.

NSRHN specifies the SRH concentration parameter for electrons (see Equation 3-215).

NSRHP specifies the SRH concentration parameter for holes (see Equation 3-216).

TAUN0 specifies SRH lifetime for electrons (see Equation 3-215).

TAUP0 specifies SRH lifetime for holes (see Equation 3-216).

VTHN specifies the thermal velocity for electrons for use in TRAP recombination.

VTHP specifies the thermaln velocity for holes for use in TRAP recombination.

Carrier Statistics Model Parameters

ASYMMETRY specifies the relative degree to which band gap narrowing applies to the conduction band versus the valence band. The value of the ASYMMETRY parameter is multiplied by the total change in band gap due to band gap narrowing and that product is applied to the conduction band edge. For example, if the ASYMMETRY parameter has a value of 1.0 then the change in band gap due to band gap narrowing is applied only to the conduction band edge and the valence band edge remains unaffected.

BGN.**E**, **BGN**.**N** and **BGN**.**C** specify the parameters of the band gap narrowing model given in Equation 3-38.

EAB specifies acceptor energy level (see Equation 3-44).

EDB specifies donor energy level (see Equation 3-43).

F.**BGN** specifies the name of a file containing a C-INTERPRETER function for the specification of temperature, composition and doping dependent bandgap narrowing models.

GCB specifies the conduction-band degeneracy factor (see Equation 3-43).

GVB specifies the valence-band degeneracy factor (see Equation 3-44).

Energy Balance Parameters

TAUMOB.**EL** specifies the relaxation time for electrons in the temperature dependent mobility model.

TAUMOB.**HO** specifies the relaxation time for holes in the temperature dependent mobility model .

TAUREL.**EL** specifies the relaxation time for electrons in the energy balance model (see Equation 3-95).

TAUREL.**HO** specifies the relaxation time for holes in the energy balance model (see Equation 3-96).

TRE.**T1**, **TRE**.**T2**, **TRE**.**T3**, **TRE**.**W1**, **TRE**.**W2**, **TRE**.**W3**, **TRH**.**T1**, **TRH**.**T2**, **TRH**.**T3**, **TRH**.**W1**, **TRH**.**W2**, and **TRH**.**W3** are used in the temperature dependent energy relaxation time model based on table data from Laux-Fischetti Monte-Carlo simulation (see Table 3-9).

Lattice Temperature Dependence Parameters

EGALPHA specifies the alpha coefficient for temperature dependence of bandgap (see Equation 3-36).

EGBETA specifies the beta coefficient for temperature dependence of bandgap (see Equation 3-36).

HC.**A**, **HC**.**B**, **HC**.**C**, and **HC**.**D** specify the values of the four coefficient of the heat capacity equation (see Equation 6-6).

LT.**TAUN** specifies the temperature dependence for electron lifetimes (see Equation 6-9).

LT.**TAUP** specifies the temperature dependence for hole lifetimes (see Equation 6-20).

POWER specifies the value of thermal power generated in a power source associated with a region in THERMAL3D.

TC.**A**, **TC**.**B**, and **TC**.**C** specify the three thermal conductivity coefficients (see Equation 6-4).

TC.**C0** specifies the equilibrium value of thermal conductivity, k(T0), in equation 13-2.

Synonym: TC.CONST

TC.**D** specifies the value of the parameter D in equation 13-5.

TC.**E** specifies the value of the parameter E in equation 13-5.

TCON.**CONST** specifies that thermal conductivity should be modeled as constant with respect to temperature. The value of the thermal conductivity is given by the value of the TC.C0 parameter.

TCON.**POWER** specifies that the temperature dependence of thermal conductivity should be modeled using equation 13-3.

TCON.**POLYNOM** specifies that the temperature dependence of thermal conductivity should be modeled using equation 13-4.

TCON.**RECIPRO** specifies that the temperature dependence of thermal conductivity should be modeled using equation 13-5.

TC.**NPOW** specifies the value of the coefficient of temperature dependence of thermal conductivity, n, in equation 13-3.

TMUN and **TMUP** specify the temperature coefficients for the temperature dependence of electron lattice mobility, and of hole lattice mobility respectively.

Oxide Material Parameters.

SEMICONDUC specifies that an oxide region is to be treated as a semiconductor.

OXCH.ONLY specifies that electron and hole concentrations are omitted from Poisson's equation in oxides.

Photogeneration Parameters.

F.**INDEX** specifies the name of a file containing a C-INTERPRETER function for the specification of wavelength dependent complex index of refraction models.

IMAG.**INDEX** specifies the imaginary portion of the refractive index of the semiconductor (see Equation 8-12). Wavelength dependent defaults exist for certain materials as documented in Appendix B.

INDEX.**FILE** specifies the filename from which refractive indices for a material are read. This parameter is useful to load wavelength dependent refractive indices for use with both mono- and multi- spectral light sources. The format of this file is:

```
<n>
wavelength(1) real index(1) imaginary index(1)
wavelength(2) real index(2) imaginary index(2)
...
wavelength(n) real index(n) imaginary index(n)
```

Where **n** is the number of lines to be read.

REAL.**INDEX** specifies the real portion of the refractive index of the semiconductor. Wavelength dependent defaults exist for certain materials as documented in Appendix B.

LASER Parameters

ALPHAA specifies the bulk absorption coeffiecient in Equation 9-2.

ALPHAR specifies the line width broadening factor in Equation 9-2.

EPSINF specifies the high frequency relative dielectric permittivity (ε_∞) (see Equation 9-2). If this parameter is not specified, it will be set equal to the static dielectric permittivity of the material.

F.**ALPHAA** specifies the name of a file containing a C-INTERPRETER function for the bulk absorption coeffiecient.

FCN parameter in the LASER free-carrier loss model.

FCP parameter in the LASER free-carrier loss model.

GAIN0 specifies the parameter g0 in Equations 9-3 and 9-11.

GAIN00 specifies the parameter g00 in Equation 9-6.

GAIN1N specifies the parameter g1n in Equation 9-6.

GAIN1P specifies the parameter g1p in Equation 9-6.

GAIN1MIN specifies the parameter g1min in Equation 9-6.

GAIN2NP specifies the parameter g2np in Equation 9-6.

GAMMA specifies the parameter in Equation 9-3. If this parameter is not specified, it will be calculated using Equation 9-5.

Material Coefficient Definition Examples

Numbered region

This example specifies SRH lifetimes and concentration independent low-field mobilities for region number 2. All other parameters use default values and parameters in other regions are unaffected.

```
MATERIAL TAUN0=5.0E-6 TAUP0=5.0E-6 MUN=3000 MUP=500 REGION=2
```

All regions

This example defines carrier lifetimes and the refractive index for all semiconductor regions.

```
MATERIAL TAUP0=2.E-6 TAUN0=2.E-6 REAL.INDEX=3.7 \
         IMAG.INDEX=1.0E-2
```

Named Material

This shows the definition of bandgap for all InGaAs regions in the structure:

```
MATERIAL MATERIAL=InGaAs EG300=2.8
```

Note: The command `MODEL PRINT` can be used to echo back default material parameters or `MATERIAL` parameter settings to the run-time output.

Note: All materials are divided into three classes: semiconductors, insulators and conductors. See Appendix B for important information about parameters required for each material class.

MEASURE

MEASURE — extracts selected electrical data from the solution.

Note: This statement is mostly obsolete and functions are replaced by the EXTRACT statement, OUTPUT statement or PROBE statement

---

Syntax

```
MEASURE <dt> [<boundary>] [OUTFILE=<filename>]
```

Parameter	Type	Default	Units
CONTACT	Integer		
E.CRIT	Real1E-8		
ELECTRON	Logical	False	
HOLE	Logical	False	
IONIZINT	Logical	False	
LRATIO	Real	1.0	
METAL.CH	Logical	False	
N.CURRENT	Logical	False	
N.LAYER	Real	15	
N.LINES	Integer	50	
N.RESIST	Logical	False	
NET.CARR	Logical	False	
NET.CHAR	Logical	False	
OUTFILE	Character		
P.CURRENT	Logical	False	
P.RESIST	Logical	False	
REGIONS	Integer	All regions	
SUBSTR	Character		
U.AUGER	Logical	False	
U.RADIATIVE	Logical	False	
U.SRH	Logical	False	
U.TOTAL	Logical	False	
X.MIN	Real	Left of device	μm

Parameter	Type	Default	Units
X.MAX	Real	Right of device	µm
Y.MIN	Real	Top of device	µm
Y.MAX	Real	Bottom of device	µm

Description

dt is one of the data type parameters described on the following page. These parameters are used to specify the type of information to be measured.

boundary is one of the boundary parameters described on page 15-63. These parameters specify which nodes will be measured.

OUTFILE specifies a filename in which simulation results and bias information will be written.

Data Type Parameters

Net carrier concentration, charge concentration, electron concentration, or hole concentration may be integrated over a section of a device. The charge on part of an electrode can be calculated, as can the current through that part. This is useful for capacitance studies in conjunction with the difference mode of the LOAD statement. The resistance of a structure cross-section, such as a diffused line, may be calculated.

E.**CRIT** specifies the critical electric field used to calculate integration integrals.

ELECTRON extracts integrated electron concentration.

HOLE extracts integrated hole concentration.

IONIZINT enable the calculation of ionization integrals. Other integral ionization parameters will be ignored unless IONIZINT is specified.

LRATIO specifies the ratio between electric field lines used in ionization integral calculation. The value of this parameter should be set from 0.5 to 1.5.

METAL.**CH** extracts integrated charge on a contact.

N.**CURRENT** extracts n-current through an electrode.

N.**LINES** specifies the number of ionization integrals.

N.**RESIST** extracts n-resistance of a cross-section.

NET.**CARR** extracts integrated carrier concentration.

NET.**CHAR** extracts integrated net charge.

NLAYERS controls the distance from the contact where electric field lines start.

P.**CURRENT** extracts p-current through an electrode.

P.**RESIST** extracts p-resistance of a cross-section.

SUBSTR selects the "substrate" electrode for electric field lines. You do not need to specify this parameter if a "substrate" electrode has been defined in the ELECTRODE statement.

U.**AUGER** specifies that the integrated Auger recombination rate is to be extracted.

U.**RADIATIVE** specifies that the integrated radiative recombination rate is to be extracted.

U.**SRH** specifies that the integrated SRH recombination rate is to be extracted.

U.**TOTAL** specifies that the integrated total recombination rate is to be extracted.

Boundary Parameters

Boundary parameters **X**.**MIN**, **X**.**MAX**, **Y**.**MIN**, and **Y**.**MAX** define a bounding box. Only nodes falling within this bounding box are included in the integration. The default bounds are the boundaries of the entire device.

CONTACT specifies the contact number. For electrode quantities (current and metal charge) a contact must be selected. Only nodes falling within the bounds and belonging to the contact are included in the integration. When IONIZINT is specified, this is the electrode used to start electric field lines.

REGIONS specifies a particular set of regions. If REGIONS is specified, only nodes within the specified bounds that are part of a particular set of regions will be integrated.

X.**MAX** specifies the x-coordinate of the right edge of the bounding box.

X.**MIN** specifies the x coordinate of the left edge of the bounding box.

Y.**MAX** specifies the y coordinate of the top of the bounding box.

Y.**MIN** specifies the y coordinate of the bottom of the bounding box.

Resistance Example

This example extracts the resistance of a p-type line diffused into a lightly doped n-substrate. Since the p-conductivity of the substrate is negligible, the integration bounds can include the whole device.

```
MEASURE P.RESIST
```

Gate Charge Example

In this example, the charge on the lower surface of a gate electrode is integrated. There is 0.05 μm of gate oxide on the surface, which is located at y=0.

```
MEASURE METAL.CH CONT=1 X.MIN=-2.0 X.MAX=2.0 \
        Y.MAX=-0.0499 Y.MIN=-0.0501
```

Ionization Integral Example

This example shows how to extract the maximum ionization integral in the device.

```
MEASURE IONIZINT CONTACT=3 SUSTR=4 N.LINES=200 \
        NLAYERS=15 LRATIO=1.1
```

This syntax was the original implementation of ionization integrals in early ATLAS versions. It has been superceded. See the SOLVE and OUTPUT statements for the recommended approach to extract ionization integrals.

MESH

MESH — generates a mesh or reads a previously generated mesh.

Syntax

```
MESH <prev>|<new> [<output>]
```

Parameter	Type	Default	Units
ATHENA	Logical	False	
CYLINDRICAL	Logical	False	
DIAG.FLIP	Logical	True	
FLIP.Y	Logical	False	
INFILE	Character		
MASTER.IN	Logical	True	
MASTER.OUT	Logical	True	
NX	Integer		
NY	Integer		
NZ	Integer		
OUTFILE	Character		
PISCES.IN	Logical	False	
SCALEY	Integer	1	
SMOOTH.KEY	Integer		
SPACE.MULT	Real	1.0	
WIDTH	Real	1.0	

Description

prev is a set of parameters that allow the user to read a previously generated mesh type.

new is a set of parameters that allow the user to initiate the generation of a rectangular mesh.

output is a set of the parameters for saving the mesh

Parameters related to reading in an existing mesh file

CYLINDRICAL specifies that the mesh being read in contains cylindrical symmetry. Since this information is not saved in the mesh file, the CYLINDRICAL parameter must be specified each time a structure with cylindrical symmetry is loaded.

FLIP.Y reverses the sign of the y coordinate.

INFILE specifies the name of a previously generated mesh that has been saved to disk.

MASTER.IN specifies a filename from which to read mesh and doping information in the SILVACO standard structure file (SSF) format. This parameter is used to read in ATHENA or DEVEDIT structure files. Typically such files contain all `REGION`, `ELECTRODE` and `DOPING` information although ATLAS allows users to modify the structure using these statements.This parameter is true by default and is the only file format supported by Silvaco.

PISCES.IN indicates that the mesh file is in the old PISCES-II format. This is not recommended or supported by Silvaco.

SCALE.Y specifies a factor by which all x- and y-coordinates are multiplied.

SPACE.MULT is a scale factor that is applied to all specified grid spacings. This parameter can be used to produce a coarse mesh and therby reduce the simulation time.

ATHENA reads mesh and doping data generated by the ATHENA PISCES-II format file. This parameter and file format is obsolete.

Parameters Related to Creation of a New Mesh

RECTANGULAR initiates the generation of a rectangular mesh.

CYLINDRICAL specifies that the mesh contains cylindrical symmetry. Since this information is not saved in the mesh file, the CYLINDRICAL parameter must be specified each time a grid with cylindrical symmetry is loaded. Structures defined as cylindrical will be rotated 360 degress about the y-axis. The coordinate x=0 **must** be on the extreme left hand side of the structure.

DIAG.FLIP flips the diagonals in a square mesh about the center of the grid. If the parameter is negated using `^DIAG.FLIP` is specified, all diagonals will be in the same direction.

NX specifies the number of nodes in the x direction.

NY specifies the number of nodes in the y direction.

NZ specifies the number of nodes in the z direction, used in DEVICE3D or BLAZE3D only.

WIDTH specifies a scale factor to represent the unsimulated dimension for 2-D simulations. This scale factor is applied to all run time and log file outputs.

Output Parameters

OUTFILE specifies the output filename to which the mesh is written.

MASTER.OUT specifies the format of the output file. This parameter is true by default so the output file will conform to SILVACO Standard Structure File Format and can be plotted in TONYPLOT.

SMOOTH.KEY specifies a smoothing index. The digits of the index are read in reverse order and interpreted as follows:

1. Triangle smoothing. All region boundaries remain fixed.
2. Triangle smoothing. Only material boundaries are maintained.
3. Node averaging.
4. Improved triangle smoothing method. This method uses diagonal flipping to reduce the number of obtuse triangles.
5. Triangle smoothing by flipping diagonals according to electric field.

Usually option 1 is sufficient. Option 2 is useful only if a device has several regions of the same material and the border between different regions is unimportant. Option 3 is not recommended when the initial mesh is basically rectangular, such as mesh information usually obtained from SSUPREM4. Option 4 is similar to option 1, but option 4 usually creates less obtuse triangles.

Mesh Definition Example

This example initiates a rectangular mesh and stores the mesh in file MESH1.STR.

```
MESH RECTANGULAR NX=40 NY=17 OUTF=MESH1.STR
```

ATHENA Interface Example

This syntax reads in a mesh from ATHENA or DEVEDIT:

```
MESH INFILE=NMOS.STR
```

When the auto- interface feature is used in DECKBUILD the program will automatically insert the MESH statement to load the result of previous programs into ATLAS.

Note: Please see the Getting Started Chapter or the on-line examples for details of the interfaces from ATHENA or DEVEDIT to ATLAS

METHOD

METHOD — sets the numerical methods to be used to solve the equations and parameters associated with the these algorithms.

Syntax

```
METHOD <gp> <mdp>
```

Parameter	Type	Default	Units
2NDORDER	Logical	True	
ATRAP	Real	0.5	
AUTONR	Logical	False	
BIGGEST	Logical	False	
BLOCK	Logical	True	
CARRIERS	Real	2	
C.ITLIMIT	Integer	500	
C.STABIL	Real	$1.0x10^{-10}$	
C.RESID	Real	$1.0x10^{-8}$	
CLIM.DD	Real	$4.5x10^{13}$	cm^{-3}
CLIM.EB	Real	0	cm^{-3}
CLIMIT	Real	10000	
CR.TOL	Real	$5.0x10-^{18}$	
CUR.PROJ	Logical	False	
CX.TOLER	Real	$1.0x10^{-5}$	
DIRECT	Logical	False	
DT.MAX	Real	∞	
DT.MIN	Real	$1.0x10^{-25}$	s
DVLIMIT	Real	0.1	
DVMAX	Real	2.5	V
ELECTRONS	Logical	True	
EXTRAPOLATE	Logical	False	
FIX.QF	Logical	False	
GMRES	Logical	False	
GUM.INIT	Integer	15	

Parameter	Type	Default	Units
GUMITS	Integer	100	
GUMMEL	Logical	False	
HALFIMPLICIT	Logical	False	
HCIR.TOL	Real	$5.0x10^{-11}$	
HCIX.TOL	Real	$5.0x10^{-4}$	
HOLES	Logical	True	
ICCG	Logical	False	
IR.TOL	Real	$5.0x10^{-15}$	
ITLIMIT	Integer	25	
IX.TOL	Real	$2.0x10^{-5}$	
LU1CRI	Real	$3.0x10^{-3}$	
LU2CRI	Real	$3.0x10^{-2}$	
L2NORM	Logical	True	
MAXTRAPS	Integer	4	
MAX.TEMP	Real	2000	K
MEINR	Logical	False	
MIN.TEMP	Real	120	K
NEG.CONC	Logical	False	
NEWTON	Logical	True	
NBLOCKIT	Integer	15	
NRCRITER	Real	0.1	
NO.POISSON	Logical	False	
PR.TOL	Real	$1.0x10^{-26}$	
PX.TOLER	Real	$1.0x10^{-5}$	
RHSNORM	Logical	False	
RXNORM	Logical	False	
QUASI	Logical	True	
SEMIIPLICIT	Logical	False	
SINGLEPOISSON	Logical	False	
TAUTO	Logical	True	
TCR.TOL	Real	100	

Parameter	Type	Default	Units
TCX.TOL	Real		
TLR.TOL	Real	100	
TLX.TOL	Real		
TMIN.FACT	Real	0.4	
TOL.LTEMP	Real	0.001	
TOL.RELAX	Real	1	
TOL.TIME	Real	$5.0x10^{-3}$	
TRAP	Logical	True	
TSETP.INCR	Real	2.0	
V.TOL	Real	$1.0x10^{-6}$	
VSATMOD.INC	Real	0.01	
XNORM	Logical	False	
WEAK	Real	200	

Description

The METHOD statement is used to set the numerical methods for subsequent solutions. All structure and model defintions should precede the METHOD statement and all biasing conditions should follow it.Parameters on the METHOD statement are used to set the solution technique, specify options for each technique and tolerances for covergence.

Parameters to select the Solution Method

Note: Details of the different solution methods can be found in the **Numerical Techniques** Chapter

BLOCK specifies that the block Newton solution method will be used as a possible solution method in subsequent solve statements until otherwise specified.The BLOCK method only has meaning when either lattice heating or energy balance is included in the simulation. For isothermal drift diffusion simulations BLOCK is equivalent to NEWTON.

BICGST switches from the default ILUCGS iterative solver to the BICGST interative solver for 3-D simulations.

DIRECT specifies that a direct linear solver should be used to solve the linear problem during 3-D simulation. By default the ILUCGS iterative solver is used for 3-D problems.

GMRES switches from the default ILUCGS iterative solver to the GMRES interative solver for 3-D simulations.

GUMMEL specifies the Gummel method will be used as a solution method in subsequent SOLVE statements until otherwise specified. If other methods (BLOCK or NEWTON) are specified in the same METHOD statement, each solution method will be applied in succession until convergence is obtained. The order that the solution methods will be applied is GUMMEL then BLOCK then NEWTON. If no solution methods are specified NEWTON is applied by default.

HALFIMPLICIT specifies that a semi-implicit scheme will be used for transient solutions in 3-D. In most cases this method is significantly less time consuming that the default TR-BDG method.

MEINR specifies the Meinerzhagens method whereby carrier temperature equations will be coupled with the associated carrier continuity equation will be used during GUMMEL iterations.

NEWTON specifies that Newton's method will be used as the solution method in subsequent SOLVE statements until otherwise specified.Certain models and boundary conditions settings require that Newton's method is used.If no solution methods are specified NEWTON is applied by default.

Parameters to select which equations are solved

CARRIERS specifies the number of carrier continuity equations that solutions will be found for. Valid values are 0, 1 and 2. CARRIERS=0 implies that only Poisson's equation is solved for. CARRIERS=1 implies that only one carrier solution will be obtained. When this is specified one should also specify either HOLES or ELECTRONS. CARRIERS=2 implies that solutions will be obtained for both electrons and holes.

ELECTRONS specifies that only electrons will be simulated for single carrier simulation. .

HOLES specifies that only holes will be simulated for single carrier simulations.

Solution Tolerance Parameters

Note: Generally the solution tolerances should not be changed. Convergence problems should be tackled by improving the mesh or checking the model and method combinations. The **Getting Started** Chapter has useful hints.

The default convergence criteria used in ATLAS consists of a combination of relative and absolute values. The program will converge if either criterion is met. This is particularly useful when low-carrier concentrations would not converge using just relative criteria.

Current convergence criteria are also used. Terminal currents are monitored at each iteration and overall convergence is allowed if currents converge along with absolute potential error.

CR.**TOL** specifies an absolute tolerance for the continuity equation.

CX.**TOL** or **C.TOL** is the relative tolerance for the continuity equation. The XNORM parameter uses parameters CX.TOL and PX.TOL to calculate convergence criteria.

HCIR.**TOL** is the absolute current convergence criteria for energy transport models.

HCIX.**TOL** is the relative current convergence criteria for energy transport models.

IR.**TOL** specifies absolute current convergence criteria.

IX.**TOL** specifies relative current convergence criteria.

PR.**TOL** specifies an absolute tolerance for the Poisson equation.

PX.**TOL** is the relative tolerance for the potential equation. The XNORM parameter uses parameters CX.TOL and PX.TOL to calculate convergence criteria.

RHSNORM specifies that only absolute errors will be used to determine convergence. If RHSNORM is selected Poisson error are measured in C/μm and the continuity error is measured in A/μm.

RXNORM specifies that both relative and absolute convergence criteria will be used in the solution method. This is the equivalent of specifying both XNORM and RHSNORM. This is the default and it is not recommended to change this.

TCR.**TOL** specifies the absolute (RHS norm) tolerance for convergence of the carrier temperature equations.

TCX.**TOL** specifies the relative (X norm) tolerance for convergence of the carrier temperature equations.

TOL.**TIME** specifies maximum local truncation error for transient simulations.

TOL.**LTEMP** specifies the temperature convergence tolerance in block iterations using the lattice heat equation.

TOL.RELAX specifies a relaxation factor for all six Poisson, continuity, and current convergence parameters (PX.TOL, CX.TOL, PR.TOL, CR.TOL, IX.TOL, and IR.TOL).

TLR.TOL specifies the relative (X norm) tolerance for convergence of the lattice temperature equation.

TLX.TOL specifies the relative (X norm) tolerance for convergence of the lattice temperature equation.

WEAK specifies the multiplication factor for weaker convergence tolerances applied when current convergence is obtained.

XNORM specifies that only the relative errors will be used to determine convergence for the drift-diffusion equations. If XNORM is used Poisson updates are measured in kT/q. and carrier updates are measured relative to the local carrier concentration.

General Parameters

ATRAP specifies the multiplication factor which reduces the electrode bias steps when a solution starts to diverge. This parameter has no effect unless the TRAP parameter is specified.

CLIM.DD is analogous to CLIMIT except it is expressed in a dimensional value representing the minimum carrier concentration that can be resolved.

CLIM.EB can be treated as a regularization parameter for the case of very small carrier concentrations for energy balance simulation. It specifies the minimum value of carrier concentration for which the relaxation term in the energy balance equation will still be properly resolved. Carrier temperatures for points where the concentration is much less than CLIM.EB, will tend to the lattice temperature.

CLIMIT specifies a concentration normalization factor. See the Numerical Techniques chapter for a complete description.

CUR.PROJ enables the use of projection method for initial guesses with current boundary conditions.

FIX.QF fixes the quasi-Fermi potential of each non-solved for carrier to a single value, instead of picking a value based on local bias.

ITLIMIT or **GITLIMIT** specifies the maximum number of allowed outer loops (Newton loops or Gummel continuity iterations).

MAXTRAPS specifies the number of times the trap procedure will be repeated in case of divergence. The value of MAXTRAPS may range from 1 to 10.

MIN.TEMP and **MAX.TEMP** are specified to control the absolute range of lattice temperatures allowed during Gummel loop iterations with lattice temperature. These parameters help insure that lattice temperatures converge during the outer loop iterations.

NBLOCKIT specifies the maximum number of BLOCK iterations. If METHOD BLOCK NEWTON is specified the solver will switch to Newton's method after NBLOCKIT Block-Newton iterations.

NEG.CONC flag allows negative carrier concentrations.

NO.POISSON flag allows omission of Poisson's equation in solution.

TMIN.FACT specifies the minimum electron or hole temperature allowable during non-linear iteration updates. TMIN.FACT is normalized to 300K.

TRAP specifies that if a solution process starts to diverge, the electrode bias steps taken from the initial approximation are reduced by the multiplication factor ATRAP.

VSATMOD.INC specifies that the derivatives of the negative differential mobility (MODEL FLDMOB EVSATMOD=1) will not be included into the Jacobian until the norm of Newton update for potential is less than the value specified by VSATMOD.INC. This is useful since the nagative differential mobility model is highly nonlinear and causes numerical stability problems.

Gummel Parameters

DVLIMIT limits the maximum potential update for a single loop.

GUM.INIT specifies the maximum number of Gummel iterations in order to obtain an initial approximation for successive Newton iterations. This parameter is used when `METHOD GUMMEL NEWTON` is specified

GUMITS specifies the maximum number of Gummel iterations.

LU1CRI, **LU2CRI** specifies amount of work per Poisson loop. The inner norm is required to (1) decrease by at least `LU1CRI` before returning, or (2) reach a factor of `LU2CRI` below the projected Newton error; whichever is smaller. If the inner norm is exceeds the projected Newton error, quadratic convergence is lost.

SINGLEPOISSON specifies that only a single Poisson iteration is to be performed per Gummel loop. In the default state, the continuity equation is only treated after the Poisson iteration has fully converged. This technique is useful where the carrier concentration and potential are strongly coupled but the initial guess is poor precluding the use of `NEWTON`.

Newton Parameters

2NDORDER specifies that second-order discretization will be used when transient simulations are performed.

AUTONR implements an automated Newton-Richardson procedure which attempts to reduce the number of LU decompositions per bias point. We strongly recommend that you use this parameter to increase the speed of `NEWTON` solutions.Iterations using `AUTONR` will appear annotated with an A in the run-time output. Often an extra iteration is added when using this parameters since the final iteration of any converged solution cannot be done using `AUTONR`.

DT.MAX specifies the maximum time-step for transient simulation.

DT.MIN specifies the minimum time-step for transient simulations.

DVMAX sets the maximum allowed potential update per Newton iteration. Large voltage steps are often required when simulating high voltage devices. If any simulation requires voltage steps of 10V or more, set `DVMAX` to 100,000. Reducing `DVMAX` may serve to damp oscillations in solutions in some cases leading to more robust behavior. However excessive reduction in `DVMAX` is not recommended since the maximum voltage step allowed will be limited by `DVMAX*ITLIMIT` .

EXTRAPOLATE specifies the use of second-order extrapolation to compute initial estimates for successive time-steps for transient simulations.

L2NORM specifies the use of L2 error norms rather than infinity norms when calculating time steps for transient simulations.

NRCRITER specifies the ratio by which the norm from the previous Newton loop must decrease in order to be able to use the same Jacobian (LU decomposition) for the current Newton loop.

TAUTO selects automatic, adaptive timesteps for transient simulations from local truncation error estimates. Automatic time-stepping is the default for second-order discretization, but is not allowed for first-order.

TSTEP.INCR specifies the maximum allowable ratio between the time step sizes of successive (increasing) time steps during transient simulation.

QUASI specifies a quasistatic approximation for transient simulations. This is useful in simulating transient simulations with long timescales where the device is in equilibrium at each timestep.

Numerical Method Defintion Example

The default numerical method is the equivalent of:

```
METHOD NEWTON CARRIERS=2
```

For more complex problems including those involving floating regions the following is recommended

```
METHOD GUMMEL NEWTON GUM.INIT=5
```

When impact ionization is combined with floating regions as in SOI or guard ring breakdown simualation the above syntax can also be used. However quicker solutions can be obtained using the SINGLEPOISSON technique:

```
METHOD GUMMEL NEWTON GUM.INIT=5 SINGLE
```

TRAP Parameter Example

This example illustrates the trap feature (often used to capture knees of I-V curves for junction breakdown).

The first SOLVE statement solves for the initial, zero bias case. In the second SOLVE statement, we attempt to solve for V2=3 volts and V3=5 volts. If such a large bias change caused the solution algorithms to diverge for this bias point, the bias steps would be multiplied by ATRAP (0.5).

An intermediate point (V2=1.5 volts, V3=2.5 volts) would be attempted before trying to obtain V2=3 volts and V3=5 volts again. If the intermediate point can not be solved for either case, then the program will continue to reduce the bias step (the next would be V2=0.75 volts and V3=1.25 volts) up to MAXTRAPS times.

```
METHOD TRAP ATRAP=0.5

SOLVE INIT

SOLVE V2=3 V3=5 OUTFILE=OUTA
```

Transient Method Example

In this transient simulation example, second-order discretization is used (by default), but the required LTE (10^{-3}) is smaller than the default. Because the Jacobian is exact for the second part (BDF-2) of the composite timestep, there should be very few factorizations for the BDF-2 interval when AUTONR is specified.

```
METHOD NEWTON TOL.TIME=1E-3 AUTONR
```

Note: For recommendations on METHOD parameters for different simulations, see the **Getting Started** Chapter or the on-line examples.

MOBILITY

MOBILITY — allows specification of mobility model parameters.

Syntax

```
MOBILITY [NUMBER=<n>] [REGION=<n>] [MATERIAL=<name>]
         [NAME=<region_name>] <parameters>
```

Parameter	Type	Default	Units
ACCN.SF	Real	0.87	
ACCP.SF	Real	0.87	
INVN.SF	Real	0.75	
INVP.SF	Real	0.75	
N.ANGLE	Real	0.0	Degrees
P.ANGLE	Real	0.0	Degrees
OXLEFTN	Real		microns
OXLEFTP	Real		microns
OXRIGHTN	Real		microns
OXRIGHTP	Real		microns
OXBOTTOMN	Real		microns
OXBOTTOMP	Real		microns
ALPHAN.ARORA	Real	-0.57	
ALPHAP.ARORA	Real	-0.57	
BETAN.ARORA	Real	-2.33	
BETAP.ARORA	Real	-2.33	
GAMMAN.ARORA	Real	2.546	
GAMMAP.ARORA	Real	2.546	
NCRITN.ARORA	Real	1.432×10^{17}	cm^{-3}
NCRITP.ARORA	Real	2.67×10^{17}	cm^{-3}
MU1N.ARORA	Real	88.0	$cm^2/(V \cdot s)$
MU1P.ARORA	Real	54.3	$cm^2/(V \cdot s)$
MU2N.ARORA	Real	1252.0	$cm^2/(V \cdot s)$
MU2P.ARORA	Real	407.0	$cm^2/(V \cdot s)$
ALPHAN.CAUG	Real	0.0	

Parameter	Type	Default	Units
ALPHAP.CAUG	Real	0.0	
BETAN.CAUG	Real	-2.3	
BETAP.CAUG	Real	-2.2	
DELTAN.CAUG	Real	0.73	
DELTAP.CAUG	Real	0.70	
GAMMAN.CAUG	Real	-3.8	
GAMMAP.CAUG	Real	-3.7	
NCRITN.CAUG	Real	$1.072x10^{17}$	cm^{-3}
NCRITP.CAUG	Real	$1.606x10^{17}$	cm^{-3}
MU1N.CAUG	Real	55.24	$cm^2/(V·s)$
MU1P.CAUG	Real	49.7	$cm^2/(V·s)$
MU2N.CAUG	Real	1429.23	$cm^2/(V·s)$
MU2P.CAUG	Real	479.37	$cm^2/(V·s)$
AL1N.WATT	Real	-0.16	
AL1P.WATT	Real	-0.296	
AL2N.WATT	Real	-2.17	
AL2P.WATT	Real	-1.62	
AL3N.WATT	Real	1.07	
AL3P.WATT	Real	1.02	
ETAN.WATT	Real	0.50	
ETAP.WATT	Real	0.33	
MREF1N.WATT	Real	481.0	$cm^2/(V·s)$
MREF1P.WATT	Real	92.8	$cm^2/(V·s)$
MREF2N.WATT	Real	591.0	$cm^2/(V·s)$
MREF2P.WATT	Real	124.0	$cm^2/(V·s)$
MREF3N.WATT	Real	1270.0	$cm^2/(V·s)$
MREF3P.WATT	Real	534.0	$cm^2/(V·s)$
MOD.WATT.N	Logical	False	
MOD.WATT.P	Logical	False	
EXP.WATT.N	Logical	False	

Parameter	Type	Default	Units
EXP.WATT.P	Logical	False	
XMINN.WATT	Real	-1.0x32	microns
XMAXN.WATT	Real	1.0x32	microns
YMAXN.WATT	Real	-1.0x32	microns
XMINP.WATT	Real	-1.0x32	microns
XMAXP.WATT	Real	1.0x32	microns
YMAXP.WATT	Real	-1.0x32	microns
YCHARN.WATT	Real	1.0x32	microns
YCHARP.WATT	Real	1.0x32	microns
ETAN	Real		
AN.CCS	Real	$4.61x10^{17}$	cm^{-3}
AP.CCS	Real	$1.0x10^{17}$	cm^{-3}
BN.CCS	Real	$1.52x10^{15}$	cm^{-3}
BP.CCS	Real	$6.25x10^{14}$	cm^{-3}
ANALYTIC.N	Logical	False	
ANALYTIC.P	Logical	False	
ARORA.N	Logical	False	
ARORA.P	Logical	False	
CONMOB.N	Logical	False	
CONMOB.P	Logical	False	
CCSMOB.N	Logical	False	
CCSMOB.P	Logical	False	
FLDMOB.N	Logical	False	
FLDMOB.P	Logical	False	
SCHWARZ.N	Logical	False	
SCHWARZ.P	Logical	False	
TASCH.N	Logical	False	
TASCH.P	Logical	False	
MOBMOD.N	Real	1	
MOBMOD.P	Real	1	
OLDSURF.N	Logical	False	

Parameter	Type	Default	Units
OLDSURF.P	Logical	False	
SURFMOB.N	Logical	False	
SURFMOB.P	Logical	False	
ASN.YAMA	Real	$1.54x10^{-5}$	cm/V
ASP.YAMA	Real	$5.35x10^{-5}$	cm/V
GN.YAMA	Real	8.8	cm/s
GP.YAMA	Real	1.6	cm/s
NREFN.YAMA	Real	$3.0x10^{16}$	cm^{-3}
NREFP.YAMA	Real	$4.0x10^{16}$	cm^{-3}
MULN.YAMA	Real	1400.0	$cm^2/(V·s)$
MULP.YAMA	Real	480.0	$cm^2/(V·s)$
SN.YAMA	Real	350	
SP.YAMA	Real	81.0	
VSN.YAMA	Real	$1.036x10^7$	cm/s
VSP.YAMA	Real	$1.200x10^7$	cm/s
ULN.YAMA	Real	$4.9x10^6$	cm/s
ULP.YAMA	Real	$2.928x10^6$	cm/s
ALPHN.CVT	Real	0.680	
ALPHP.CVT	Real	0.71	
BN.CVT	Real	$4.75x10^7$	cm/(K·s)
BP.CVT	Real	$9.925x10^6$	cm/(K·s)
BETAN.CVT	Real	2.00	
BETAP.CVT	Real	2.00	
CN.CVT	Real	$1.74x10^5$	
CP.CVT	Real	$8.842x10^5$	
CRN.CVT	Real	$9.68x10^{16}$	cm^{-3}
CRP.CVT	Real	$2.23x10^{17}$	cm^{-3}
CSN.CVT	Real	$3.43x10^{20}$	cm^{-3}
CSP.CVT	Real	$6.10x10^{20}$	cm^{-3}

Parameter	Type	Default	Units
DELN.CVT	Real	$5.82x10^{14}$	V/s
DELP.CVT	Real	$2.0546x10^{14}$	V/s
GAMN.CVT	Real	2.5	
GAMP.CVT	Real	2.2	
MU0N.CVT	Real	52.2	$cm^2/(V \cdot s)$
MU0P.CVT	Real	44.9	$cm^2/(V \cdot s)$
MU1N.CVT	Real	43.4	$cm^2/(V \cdot s)$
MU1P.CVT	Real	29.0	$cm^2/(V \cdot s)$
MUMAXN.CVT	Real	1417.0	$cm^2/(V \cdot s)$
MUMAXP.CVT	Real	470.5	$cm^2/(V \cdot s)$
PCN.CVT	Real	0.0	cm^{-3}
PCP.CVT	Real	$0.23x10^{16}$	cm^{-3}
TAUN.CVT	Real	0.125	
TAUP.CVT	Real	0.0317	
DEVICE	Character		
MATERIAL	Character		
NAME	Character		
ECRITN	Real	$4.0x10^3$	V/cm
ECRITP	Real	$4.0x10^3$	V/cm
GAMMAN	Real	4.0	
GAMMAP	Real	1.0	
RN.TAS	Real	2	
RP.TAS	Real	3	
MUBN.TAS	Real	1150	
MUBP.TAS	Real	270	
TMUBN.TAS	Real	2.5	
TMUBP.TAS	Real	1.4	
DN.TAS	Real	$3.2x10^{-9}$	
DP.TAS	Real	$2.35x10^{-9}$	
P1N.TAS	Real	0.09	

Parameter	Type	Default	Units
P1P.TAS	Real	0.334	
B1N.TAS	Real	1.75	
B1P.TAS	Real	1.5	
P2N.TAS	Real	$4.53x10^{-8}$	
P2P.TAS	Real	$3.14x10^{-7}$	
B2N.TAS	Real	-0.25	
B2P.TAS	Real	-0.3	
Z11N.TAS	Real	0.0388	
Z11P.TAS	Real	0.039	
Z22N.TAS	Real	$1.73x10^{-5}$	
Z22P.TAS	Real	$1.51x10^{-5}$	
ESRN.TAS	Real	$2.449x10^{7}$	
ESRP.TAS	Real	$1.0x10^{9}$	
BETAN.TAS	Real	2	
BETAP.TAS	Real	1	
N2N.TAS	Real	$1.1x10^{21}$	
N2P.TAS	Real	$1.4x10^{18}$	
N1N.TAS	Real	$2.0x10^{19}$	
N1P.TAS	Real	$8.4x10^{16}$	
ALPHAN.TAS	Real	2	
ALPHAP.TAS	Real	3.4	
BETAN	Real	2.0	
BETAP	Real	1.0	
VSATN	Real		cm/s
VSATP	Real		cm/s
ALPHAN.FLD	Real	$2.4x10^{7}$	cm/s
ALPHAP.FLD	Real	$2.4x10^{7}$	cm/s
THETAN.FLD	Real	0.8	
THETAP.FLD	Real	0.8	
TNOMN.FLD	Real	600.0	K

Parameter	Type	Default	Units
TNOMP.FLD	Real	600.0	K
EVSATMOD	Real	0	
HVSATMOD	Real	0	
F.CONMUN	Character		
F.CONMUP	Character		
F.MUNSAT	Character		
F.MUPSAT	Character		
F.VSATN	Character		
F.VSATP	Character		
MUDEG.N	Real	1.0	
MUDEG.P	Real	1.0	
MUN	Real	1.0	$cm^2/(V \cdot s)$
MUP	Real	1.0	$cm^2/(V \cdot s)$
TMUP	Real	1.0	
TMUN	Real	1.0	
KLA.N	Logical	False	
KLA.P	Logical	False	
SHI.N	Logical	False	
SHI.P	Logical	False	
MUMAXN.KLA	Real	1417.0	$cm^2/(V \cdot s)$
MUMAXP.KLA	Real	470.5.0	$cm^2/(V \cdot s)$
MUMINN.KLA	Real	52.2	$cm^2/(V \cdot s)$
MUMINP.KLA	Real	44.9	$cm^2/(V \cdot s)$
NREF1N.KLA	Real	9.68x16	cm^{-3}
NREF1P.KLA	Real	2.23x17	cm^{-3}
ALPHA1N.KLA	Real	0.68	
ALPHA1P.KLA	Real	0.719	
THETAN.KLA	Real	2.285	
THETAP.KLA	Real	2.247	
CD.KLA	Real	0.21	
CA.KLA	Real	0.50	

Parameter	Type	Default	Units
NREFD.KLA	Real	4.0x20	cm^{-3}
NREFA.KLA	Real	7.2x20	cm^{-3}
FCW.KLA	Real	2.459	
FBH.KLA	Real	3.828	
S1.KLA	Real	0.89233	
S2.KLA	Real	0.41372	
S3.KLA	Real	0.19778	
S4.KLA	Real	0.28227	
S5.KLA	Real	0.005978	
S6.KLA	Real	1.80618	
S7.KLA	Real	0.72169	
R1.KLA	Real	0.7643	
R2.KLA	Real	2.2999	
R3.KLA	Real	6.5502	
R4.KLA	Real	2.3670	
R5.KLA	Real	-0.8552	
R6.KLA	Real	0.6478	
MU0N.SHI	Real	1430.0	$cm^2/(V \cdot s)$
MU0P.SHI	Real	500.0	$cm^2/(V \cdot s)$
E1N.SHI	Real	6.3e3	V/cm
E1P.SHI	Real	8.0e3	V/cm
E2N.SHI	Real	0.77e6	V/cm
E2P.SHI	Real	3.9e5	V/cm
P1N.SHI	Real	0.28	
P1P.SHI	Real	0.3	
P2N.SHI	Real	2.9	
P2P.SHI	Real	1.0	
THETAN.SHI	Real	2.285	
THETAP.SHI	Real	2.247	

Description

MATERIAL specifies which material from the table B-1 that the MOBILITY statement should apply to. If a material is specified, then all regions defined as being composed of that material will be affected.

DEVICE specifies the device in MIXEDMODE simulation that the MOBILITY statement should apply to.

NAME specifies the name of the region that the MOBILITY statement should apply. Note that the name must match the name specified in the NAME parameter of the REGION statement.

Mobility Model Flags

N.ANGLE specifies angle for application of electron mobility parameters in simulation of anisotropic mobility..

P.ANGLE specifies angle for application of hole mobility parameters in simulation of anisotropic mobility.

FLDMOB.**N** specifies a lateral electric field dependent model for electrons (see Equation 3-198).

FLDMOB.**P** specifies a lateral electric field dependent model for holes (see Equation 3-199).

CONMOB.**N** specifies that doping concentration dependent model to be used for electrons.

CONMOB.**P** specifies that a doping concentration dependent model is to be used for holes.

CCSMOB.**N** specifies that carrier-carrier scattering model is to be used for electrons (see Equations 3-128 - 3-131).

CCSMOB.**P** specifies that carrier-carrier scattering model is to be used for holes (see Equations 3-128 - 3-131).

ANALYTIC.**N** specifies that the analytic concentration dependent model is to be used for electrons (see Equation 3-124).

ANALYTIC.**P** specifies that the analytic concentration dependent model is to be used for holes (see Equation 3-125).

ARORA.**N** specifies that the Arora analytic concentration dependent model is to be used for electrons (see Equation 3-126).

ARORA.**P** specifies that the Arora analytic concentration dependent model is to be used for electrons (see Equation 3-127).

SURFMOB.**N** invokes the effective field based surface mobility model for electrons (see Equation 3-190).

SURFMOB.**P** invokes the effective field based surface mobility model for holes (see Equation 3-191).

SCHWARZ.N specifies the use of transverse electric field-dependent mobility models for electrons. See TFLDMB1 or SCHWARZ in the MODELS statement.

SCHWARZ.P specifies the use of transverse electric field-dependent mobility models for holes. See TFLDMB1 or SCHWARZ in the MODELS statement.

TASCH.N specifies a transverse electric field dependent mobility model for electrons based on Tasch[7,9] (see Equations 3-171 through 3-189).

TASCH.P specifies a transverse electric field dependent mobility model for holes based on Tasch[7,9] (see Equations 3-171 through 3-189).

EVSATMOD specifies which parallel field dependent mobility model (see Equations 3-198 and 5-50) should be used for electrons as follows:

1. Use the standard saturation model (Equation 3-198).
2. Use the negative differential mobility saturation model (Equation 5-50).

MOBMOD.**N** specifies transverse field degradation for electrons as follows:

1. No transverse degradation.
2. Use the Watt or Tasch transverse field models depending on the settings of FIELDMOB1 and FIELDMOB2.
3. Use the Yamaguchi transverse field dependent model.

4. Use the CVT transverse field dependent model.

MOBMOD.**P** specifies transverse field degradation for electrons as follows:

1. No transverse degradation.
2. Use the Watt or Tasch transverse field models depending on the settings of `FIELDMOB1` and `FIELDMOB2`.
3. Use the Yamaguchi transverse field dependent model.
4. Use the CVT transverse field dependent model.

KLA.**N** turns on Klaassen's mobility model for electrons (see Equations 3-132 through 3-157.

KLA.**P** turns on Klaassen's mobility model for holes (see Equations 3-132 through 3-157).

MOD.**WATT**.**N** turns on modified Watt mobility model for electrons (see page 3-194).

MOD.**WATT**.**P**turns on modified Watt mobility model for holes (see page 3-195).

EXP.**WATT**.**N** turns on exponertial modification to Watt's mobility model for electrons (see Equation 3-194).

EXP.**WATT**.**P** turns on exponential modification to Watt's mobility model for holes (see Equation 3-195).

SHI.**N** turns on Shirahata's mobility model for electrons (see Equation 3-196).

SHI.**P** turns on Shirahata's mobility model for holes (see Equation 3-197)

Temperature Dependent Low Field Mobility Parameters

MUN see Table 3-16.

MUP see Table 3-16.

TMUN see Table 3-16.

TMUP see Table 3-16.

Arora Concentration Dependent Mobility Model Parameters

MU1N.**ARORA** see Table 3-19.

MU1P.**ARORA** see Table 3-19.

MU2N.**ARORA** see Table 3-19.

MU2P.**ARORA** see Table 3-19.

ALPHAN.**ARORA** see Table 3-19.

ALPHAP.**ARORA** see Table 3-19.

BETAN.**ARORA** see Table 3-19.

BETAP.**ARORA** see Table 3-19.

GAMMAN.**ARORA** see Table 3-19.

GAMMAP.**ARORA** see Table 3-19.

NCRITN.**ARORA** see Table 3-19.

NCRITP.**ARORA** see Table 3-19.

Caughey-Thomas Concentration Dependent Model Parameters

MU1N.**CAUG** see Table 3-18.

MU1P.**CAUG** see Table 3-18.

MU2N.**CAUG** see Table 3-18.

MU2P.CAUG see Table 3-18.

ALPHAN.CAUG see Table 3-18.

ALPHAP.CAUG see Table 3-18.

BETAN.CAUG see Table 3-18.

BETAP.CAUG see Table 3-18.

GAMMAN.CAUG see Table 3-18.

GAMMAP.CAUG see Table 3-18.

DELTAN.CAUG see Table 3-18.

DELTAP.CAUG see Table 3-18.

NCRITN.CAUG see Table 3-18.

NCRITP.CAUG see Table 3-18.

Carrier-Carrier Scattering Model Parameters

AN.CCS see Table 3-20.

AP.CCS see Table 3-20.

BN.CCS see Table 3-20

BP.CCS see Table 3-20.

Parallel Field Dependent Model Parameters

BETAN see Table 3-33.

BETAP see Table 3-33.

GAMMAN see Table 5-3.

GAMMAP see Table 5-3.

ECRITN see Table 5-3.

ECRITP see Table 5-3.

VSATN see Tables 3-33 and 5-3.

VSATP see Tables 3-33 and 5-3.

ALPHAN.FLD see Table 3-33.

ALPHAP.FLD see Table 3-33.

THETAN.FLD see Table 3-33.

THETAP.FLD see Table 3-33.

TNOMN.FLD see Table 3-33.

TNOMP.FLD see Table 3-33.

Yamaguchi Transverse Field Dependent Model Parameters

SN.YAMA see Table 3-29.

SP.YAMA see Table 3-29.

NREFN.YAMA see Table 3-29.

NREFP.YAMA see Table 3-29.

MULN.YAMA see Table 3-29.

MULP.YAMA see Table3-29.

ASN.YAMA see Table 3-29.

ASP.YAMA see Table 3-29.

VSN.YAMA see Table 3-29.

VSP.YAMA see Table 3-29.

ULN.YAMA see Table 3-29.

ULP.YAMA see Table 3-29.

GN.YAMA see Table3-29.

GP.YAMA see Table 3-29.

CVT Transverse Field Dependent Model Parameters

BN.CVT see Table 3-28

BP.CVT see Table 3-28

CN.CVT see Table 3-28

CP.CVT see Table 3-28

TAUN.CVT see Table 3-28.

TAUP.CVT see Table 3-28.

GAMN.CVT see Table 3-28.

GAMP.CVT see Table 3-28.

MU0N.CVT see Table 3-28.

MU0P.CVT see Table3-28.

MU1N.CVT see Table 3-28..

MU1P.CVT see Table 3-28.

MUMAXN.CVT see Table 3-28.

MUMAXP.CVT see Table 3-28.

CRN.CVT see Table 3-28.

CRP.CVT see Table 3-28.

CSN.CVT see Table 3-28.

CSP.CVT see Table 3-28.

ALPHN.CVT see Table 3-28.

ALPHP.CVT see Table 3-28.

BETAN.CVT see Table 3-28.

BETAP.CVT see Table 3-28.

PCN.CVT see Table 3-28.

PCP.CVT see Table 3-28.

DELN.CVT see Table 3-28.

DELP.CVT see Table 3-28.

Watt Effective Transverse Field Depenedent Model Parameters

ETAN.WATT see Table 3-31

ETAP.WATT see Table 3-31

MREF1N.WATT 3-31.

MREF1P.WATT 3-31.

MREF2N.WATT see Table 3-31.

MREF2P.WATT see Table3-31.

MREF3N.WATT see Table 3-31.

MREF3P.WATT see Table 3-31.

AL1N.WATT see Table 3-31.

AL1P.WATT see Table 3-31.

AL2N.WATT see Table 3-31.

AL2P.WATT see Table 3-31.

AL3N.WATT see Table 3-31.

AL3P.WATT see Table 3-31

Tasch Mobility Model Parameters

RN.TAS see Table 3-30

RP.TAS see Table 3-30

MUBN.TAS see Table 3-30

MUBP.TAS see Table 3-30

TMUBN.TAS see Table 3-30

TMUBP.TAS see Table 3-30

DN.TAS see Table 3-30

DP.TAS see Table 3-30

P1N.TAS see Table 3-30

P1P.TAS see Table 3-30

B1N.TAS see Table 3-30

B1P.TAS see Table 3-30

P2N.TAS see Table 3-30

P2P.TAS see Table 3-30

B2N.TAS see Table 3-30

B2P.TAS see Table 3-30

Z11N.TAS see Table 3-30

Z11P.TAS see Table 3-30

Z22N.TAS see Table 3-30

Z22P.TAS see Table 3-30

ESRN.TAS see Table 3-30

ESRP.TAS see Table 3-30

BETAN.TAS see Table 3-30

BETAP.TAS see Table 3-30

N2N.TAS see Table 3-30

N2P.TAS see Table 3-30

N1N.TAS see Table 3-30

N1P.TAS see Table 3-30

ALPHAN.TAS see Table 3-30

ALPHAP.TAS see Table 3-30

Klaassen's Mobility Model Parameters

MUMAXN.KLA see Table 3-21.

MUMAXP.KLA see Table 3-21

MUMINN.KLA see Table 3-23.

MUMINP.KLA see Table 3-23.

NREF1N.KLA see Table 3-22.

NREF1P.KLA see Table 3-22.

ALPHA1N.KLA see Table 3-22.

ALPHA1P.KLA see Table 3-22.

THETAN.KLA see Table 3-21

THETAP.KLA see Table 3-21

CD.KLA see Table 3-27

CA.KLA see Table 3-27.

NREFD.KLA see Table 3-27

NREFA.KLA see Table 3-27

FCW.KLA see Table 3-26.

FBH.KLA see Table 3-26.

S1.KLA see Table 3-24.

S2.KLA see Table 3-24.

S3.KLA see Table 3-24.

S4.KLA see Table 3-24.

S5.KLA see Table 3-24.

S6.KLA see Table 3-24.

S7.KLA see Table 3-24.

R1.KLA see Table 3-25.

R2.KLA see Table 3-25.

R3.KLA see Table 3-25.

R4.**KLA** see Table 3-25.

R5.**KLA** see Table 3-25.

R6.**KLA** see Table 3-25.

The Modified Watt Mobility Model Parameters

XMINN.**WATT** see pege 3-52.

XMAXN.**WATT** see page 3-52.

YMAXN.**WATT** see page 3-52.

XMINP.**WATT** see page 3-52.

XMAXP.**WATT** see page 3-52.

YMAXP.**WATT** see page 3-52.

YCHARN.**WATT** see page 3-52.

YCHARP.**WATT** see page 3-52.

Shirahata's Mobility Model Parameters

MU0N.**SHI** see Table 3-32.

MU0P.**SHI** see Table 3-32.

E1N.**SHI** see Table 3-32.

E1P.**SHI** see Table 3-32.

E2N.**SHI** see Table 3-32.

E2P.**SHI** see Table 3-32.

P1N.**SHI** see Table 3-32.

P1P.**SHI** see Table 3-32.

P2N.**SHI** see Table 3-32.

P2P.**SHI** see Table 3-32.

THETAN.**SHI** see Table 3-32.

THETAP.**SHI** see Table 3-32.

SCHWARZ and TASCH Transverse Field Dependent Model Parameters

ACCN.**SF** specifies the accumulation saturation factor which describes the ratio of the electron concentration in the accumulation layer before and after bending of conductivity and valence bands for electron mobility.

ACCN.**SF** specifies the accumulation saturation factor which describes the ratio of the hole concentration in the accumulation layer before and after bending of conductivity and valence bands for hole mobility.

INVN.**SF** specifies the inversion saturation factor which describes the ratio of the electron concentration in the inversion layer before and after the bending of conductivity and valence bands for electron mobility.

INVN.**SF** specifies the inversion saturation factor which describes the ratio of the hole concentration in the inversion layer before and after the bending of conductivity and valence bands for hole mobility.

OXBOTTOMN specifies the coordinate of the bottom edge of the gate oxide for a MOSFET transistor for electron mobility.

OXBOTTOMP specifies the coordinate of the bottom edge of the gate oxide for a MOSFET transistor for hole mobility.

OXLEFTN specifies the coordinate of the left edge of the gate oxide for a MOSFET transistor for electron mobility.

OXLEFTP specifies the coordinate of the left edge of the gate oxide for a MOSFET transistor for hole mobility.

OXRIGHTN specifies the coordinate of the right edge of the gate oxide for a MOSFET transistor for electron mobility.

OXRIGHTP specifies the coordinate of the right edge of the gate oxide for a MOSFET transistor for hole mobility.

Interpreter Functions

F.CONMUN specifies the name of a file containing a C-INTERPRETER function for the specification of temperature, composition and doping dependent electron mobility model.

F.CONMUP specifies the name of a file containing a C-INTERPRETER function for the specification of temperature, composition and doping dependent hole mobility model.

F.MUNSAT specifies the name of a file containing a C-INTERPRETER function for the specification of parallel field dependent electron mobility model for velocity saturation.

F.MUPSAT specifies the name of a file containing a C-INTERPRETER function for the specification of parallel field dependent hole mobility model for velocity saturation.

F.VSATN specifies the name of a file containing a C-INTERPRETER function for the specification of temperature and composition dependent electron saturation velocity models.

F.VSATP specifies the name of a file containing a C-INTERPRETER function for the specification of temperature and composition dependent electron saturation velocity models.

Example Selecting the Modifed Watt Model

The following example set the Modified Watt Surface mobility model for MOSFETs. This is the best model for planar MOS simulation. The MOBILITY statement is used to set the models and to specify the value of the depth of action of the Modified Watt model.

```
MODELS CONMOB FLDMOB SRH  MIN.SURF  PRINT
MOBILITY WATT.N MOD.WATT.N YMAXN.WATT=0.01
```

MODELS

MODELS — specifies model flags to indicate the inclusion of various physical mechanisms and models.

Various other parameters such as the global temperature for the simulation.

Syntax

```
MODELS <mf> [<gp>] [<mdp>
```

Parameter	Type	Default	Units
A.TEMP	Logical	False	
ACC.SF	Real	0.87	
ALN1	Real	-0.160	
ALN2	Real	-2.17	
ALN3	Real	1.07	
ALP1	Real	-0.296	
ALP2	Real	-1.62	
ALP3	Real	1.02	
ANALYTIC	Logical	False	
AR.MU1N	Real	88	
AR.MU2N	Real	1252	
AR.MU1P	Real	54.3	
AR.MU2P	Real	407	
ARORA	Logical	False	
AUGER	Logical	False	
AUGGEN	Logical	False	
BB.A	Real	$4.0x10^{14}$	$cm^{-1/2}V^{-5/2}s^{-1}$
BB.B	Real	$1.97x10^{7}$	$cm^{-1/2}V^{-2}s^{-1}$
BB.GAMMA	Real	2.5	V/cm
BBT.KL	Logical	False	
BBT.STD	Logical	False	
BGN	Logical	False	
BGN.SLOTBOOM	Logical	False	
BIPOLAR	Logical	False	
BOLTZMANN	Logical	True	
B.ELECTRONS	Real	2	

Parameter	Type	Default	Units
B.HOLES	Real	1	
C0	Real	$2.5x10^{-10}$	
CALC.FERMI	Logical	False	
CAVITY.LENGTH	Real		μm
CCS.EA	Real	$4.61x10^{17}$	
CCS.EB	Real	$1.52x10^{15}$	
CCS.HA	Real	$1.0x10^{17}$	
CCS.HB	Real	$6.25x10^{14}$	
CCSMOB	Logical	False	
CGATE.N	Real	0.025	
CGATE.P	Real	150.0	
CHIA	Real	3e5	
CHIB	Real	$5.0x10^{4}$	
CHI.HOLES	Real	$4.6x10^{4}$	
CONMOB	Logical	False	
CONSRH	Logical	False	
CVT	Logical	False	
B.DORT	Real	1.0	
D.DORT	Real	$2.5x10^{6}$	
N.DORT	Logical	False	
P.DORT	Logical	False	
DEVDEG.E	Logical	False	
DEVDEG.H	Logical	False	
DEVDEG.B	Logical	False	
DEVICE	Character		
DRIFT.DIFF	Logical	True	
E.BENDING	Logical	False	
E.TAUR.VAR	Logical	False	
ECRIT	Real	4000.0	V/cm
EIGENS	Integer	15	
ENERGY.STEP	Real	0.0025	eV

Parameter	Type	Default	Units
ERASE	Logical	False	
ETAN	Real	0.50	
ETAP	Real	0.33	
EVSATMOD	Integer	0	
FERMIDIRAC	Logical	False	
FIXED.FERMI	Logical	False	
FLDMOB	Logical	False	
F.AE	Real	$1.82x10^{-7}$	$CV^{-2}s^{-1}$
F.BE	Real	$1.90x10^{8}$	V/cm
F.AH	Real	$1.82x10^{-7}$	$CV^{-2}s^{-1}$
F.BH	Real	$1.90x10^{8}$	V/cm
F.KSN	Character		
F.KSP	Character		
FNORD	Logical	False	
FNPP	Logical	False	
GAINMOD	Logical	0	
HANSCHQM	Logical	False	
H.BENDING	Logical	False	
H.TAUR.VAR	Logical	False	
HCTE	Logical	False	
HCTE.EL	Logical	False	
HCTE.HO	Logical	False	
HEAT.FULL	Logical	False	
HEAT.PETHOM	Logical	False	
HEI	Logical	False	
HHI	Logical	False	
HVSATMOD	Integer	0	
IG.ELINR	Real	$6.16x10^{-6}$	cm
IG.HLINR	Real	$6.16x10^{-6}$	cm
IG.ELINF	Real	$9.2x10^{-7}$	cm
IG.HLINF	Real	$9.2x10^{-7}$	cm

Parameter	Type	Default	Units
IG.EBETA	Real	$2.59x10^{-4}$	$(V.cm)^{0.5}$
IG.HBETA	Real	$2.59x10^{-4}$	$(V.cm)^{0.5}$
IG.EETA	Real	$2.0x10^{-5}$	$V^{1/3}.cm^{2/3}$
IG.HETA	Real	$2.0x10^{-5}$	$V^{1/3}.cm^{2/3}$
IG.EEOX	Real	$9.0x10^{4}$	V/cm
IG.HEOX	Real	$9.0x10^{4}$	V/cm
IG.EB0	Real	3.2	V
IG.HB0	Real	4.0	V
IMPACT	Logical	False	
INCOMPLETE	Logical	False	
INFINITY	Real	0.001	
IONIZ	Logical	False	
INV.SF	Real	0.75	
KSN	Integer	0-1	
KSP	Integer	0-1	
KLA	Logical	False	
KAUGDP	Real	0.72	
L.TEMP	Logical	False	
LAT.TEMP	Logical	False	
LASER	Logical	False	
LMODE	Logical	False	
LAS.ABSORPTION	Logical	False	
LAS.EINIT	Real		eV
LAS.ESEP	Real		eV
LAS.EFINAL	Real		eV
LAS.FCARRIER	Logical	False	
LAS.ITMAX	Integer	30	
LAS.LOSSES	Real	0	cm^{-1}
LAS.MAXCH	Real	2.5	
LAS.MIRROR	Real	90	%
LAS.MULTISAVE	Logical	False	

Parameter	Type	Default	Units
LAS.NEFF	Real	3.57	
LAS.OMEGA	Real	$2.16x10^{15}$	1/sec
LAS.SIN	Real	$1.0x10^{4}$	cm^{-1}
LAS.SPECSAVE	Integer	1	
LAS.TAUSS	Real	0.05	
LAS.TIMERATE	Logical	True	
LAS.TOLER	Real	0.001	
MATERIAL	Character		
MIN.SURF	Logical	False	
MOBMOD	Integer		
MOS	Logical	False	
MREFN1	Real	481.0	cm^2/Vs
MREFN2	Real	591.0	cm^2/Vs
MREFN3	Real	1270.0	cm^2/Vs
MREFP1	Real	92.8	cm^2/Vs
MREFP2	Real	124.	cm^2/Vs
MREFP3	Real	534.	cm^2/Vs
N.CONCANNON	Logical	False	
N.TEMP	Logical	False	
NAME	Character		
NEARFLG	Logical	False	
OPTR	Logical	False	
OX.LEFT	Real		mm
OX.RIGHT	Real		µm
OX.BOTTOM	Real	0.0	µm
P.CONCANNON	Logical	False	
P.TEMP	Logical	False	
PATH.N	Real	$3.4x10^{7}$	microns
PATH.P	Real	$2.38x10^{-7}$	microns
PEFF.N	Real	3.2	eV

Parameter	Type	Default	Units
PEFF.P	Real	4.8	eV
PHOTON.ENERGY	Real		eV
POST.SCHRO	Logical	False	
PRINT	Logical	False	
PROGRAM	Logical	False	
QUANTUM	Logical	False	
QX.MIN	Real	LHS of device	µm
QX.MAX	Real	RHS of device	µm
SCHRO	Logical	False	
SI20	Logical	False	
SHI	Logical	False	
SPEC.NAME	Character	Spectrum.log	
SRH	Logical	False	
SURFMOB	Logical	False	
TATUN	Logical	False	
TAUMOB	Logical	False	
TAUTEM	Logical	False	
TEMPERATURE	Real	300	K
TFLDMB1	Logical	False	
TFLDMB2	Logical	False	
THETA.N	Real	60.0	degrees
THETA.P	Real	60.0	degrees
YAMAGUCHI	Logical	False	

Description

mf is one or more of the model flags described below. Model flags are separated into mobility, recombination, generation, energy balance and lattice heating sections.

gp is one or more of the general parameters described on page 15-97. These parameters are used to specify general information used during the simulation.

mdp is one or more of the model dependent parameters described on page 15-100.

Mobility Model Flags

Note: See the MOBILITY statement for alternative ways to set mobility models

ANALYTIC specifies an analytic concentration dependent mobility model for silicon which includes temperature dependence (see Equations 3-124 and 3-125).

ARORA specifies an analytic concentration and temperature dependent model according to Arora (see Equations 3-126 and 3-127).

CCSMOB specifies carrier-carrier scattering model according to Dorkel and Leturq (see Equation 3-128- 3-131).

CONMOB specifies that a concentration dependent mobility model be used for silicon and gallium arsenide. This model is a doping versus mobility table valid for 300K only.

CVT specifies that the CVT transverse field dependent mobility model is used for the simulation.

FLDMOB specifies a lateral electric field-dependent model (see Equations 3-198 and 5-50). The EVSATMOD parameter may be used to define which field dependent equation is used..

KLA specifies that the Klaassen mobility model (See Equations 3-132 - 3-157) will be used for electrons and holes.

MIN.SURF specifies that the WATT, TASCH, or SHI mobility models should only apply to minority carriers.

MOBMOD specifies mobility degradation by longitudinal electric field only (MOBMOD=1), or by both longitudinal and transverse electric fields (MOBMOD=2). When MOBMOD=2, the parameters ACC.SF, INV.SF, OX.BOTTOM, OX.LEFT, and OX.RIGHT.This parameter is only used with TFLDMB1 and TFLDMB2

SHI specifies that the Shirahata mobility model (See Equation 3-197) will be used for electrons and holes.

SURFMOB or **WATT** invokes the effective field based surface mobility model (see Equations 3-190 and 3-191). SURFMOB parameters are used in the calculation of surface mobility according to the J.T. Watt model[5]. Do not specify this parameter unless S-PISCES is installed on your system.

TFLDMB1 or **SCHWARZ** specifies the use of transverse electric field-dependent mobility models. The electron model is based on the Schwarz and Russek equations [6] and is implemented in [7]. The hole model, which is used only when MOBMOD=2, is based on the Watt and Plummer equations[8].

TFLDMB2 or **TASCH** specifies a transverse electric field dependent mobility model for electrons and holes based on a semi-empirical equation [7,9].

YAMAGUCHI specifies that the Yamaguchi transverse field dependent mobility model is used in the simulation.

Recombination Model Flags

AUGER specifies Auger recombination (see Equation 3-227)

AUGGEN specifies that the Auger recombination model will be used as a generation term as well as a recombination term.

CONSRH Specifies Shockley-Read-Hall recombination using concentration dependent lifetimes (see Equations 3-215 and 3-216).

KLAAUG enables Klaassen's model for concentration dependent auger coefficients.

KLASRH enables Klaassen's model for concentration dependent lifetimes for Shockley Read Hall recombination.

OPTR selects the optical recombination model (see Equation 3-226). When this parameter is specified, the COPT parameter of the MATERIAL statement should be specified.

SRH specifies Shockley-Read-Hall recombination using fixed lifetimes (see Equation 3-213).

TATUN enables the Klaassen trap-assisted tunneling model.

Generation Model Flags

BBT.KL specifies a band-to-band tunneling model according to Klaassen.

BBT.STD specifies a standard band-to-band tunneling model (see Equation 3-270).

DEVDEG.E activates the device degradation caused by hot electron injection current. HEI should also be specified in this case.

DEVDEG.H activates the device degradation caused by hot hole injection current. HHI should also be specified in this case.

DEVDEG.B activates the device degradation caused by both the hot electron and hot hole injection currents. HEI and HHI should also be specified in this case.

Note: The specification of either DEVDEG.E, DEVDEG.H, or DEVDEG.B will initialize the density of electron (hole) like traps on the interface and will clear out the trapped electron (hole) density on the interface if any.

E.BENDING specifies that electron band bending will be taken into account for electron injection (see Equation 3-279).

FNORD selects a self-consistent Fowler-Nordheim tunneling model (see Equation 3-274).This is the recommended approach for calculating Fowler-Nordheim current.

FNPP selects a post-processing Fowler-Nordheim tunneling model (see Equation 3-275). Generally the Fowler-Nordheim current does not cause convergence problems so this appraoch is not required.

H.BENDING specifies that hole band bending will be taken into account for hole injection (see Equation 3-282).

HEI specifies the calculation of hot electron injection into oxides. This parameter can be used to simulate MOS gate current or EPROM programming. In transient mode, the oxide current is self-consistently added to the floating gate charge. See Equation 3-277.

HHI specifies the calculation of hot hole current into an oxide in a similar manner to HEI (see Equation 3-278).

IMPACT invokes the empirical impact ionization model with ionization coefficients taken from [12]. More rigorous impact ionization models may be specified with the IMPACT statement.

NEARFLG specifies the model used for oxide transport when HEI or HHI are used. The default is false which sets a purely drift based model assigning gate current to the electrodes where the electric field lines through the oxide terminate. Setting NEARFLG replaces this model with one assuming the gate current is flowing the electrode physically nearest the point of injection. This assumes a purely diffusion transport mechanism.Setting NEARFLG for devices with only one gate or a coarse mesh is advised.

N.CONCANNON enables the Concannon gate current model for electrons.

P.CONCANNON enables the Concannon gate current model for holes.

Classical Carrier Statistics Model Flags

BGN specifies band-gap narrowing (see Equation 3-38).

BGN.SLOTBOOM specifies band-gap narrowing using the original Slotboom coefficients (see Equation 3-38).

BOLTZMANN specifies that Boltzmann carrier statistics be used (see Equations 3-25 and 3-26).

FERMIDIRAC specifies that Fermi-Dirac carrier statistics be used (see Equations 3-25 – 3-62).

INCOMPLETE accounts for incomplete ionization of impurities in Fermi-Dirac statistics (see Equations 3-43 and 3-44).

IONIZ accounts for complete ionization of heavily doped silicon when using INCOMPLETE.

Quantum Carrier Statistics Model Flags

EIGENS specifies the number of eigenstates solved for by the Poisson-Schrodinger solver.

CALC.FERMI specifies that the Fermi level used in calculation of the quantum carrier concentration in the Schrodinger - Poisson model is calculated from the classic carrier concentrations.

FIXED.FERMI specifies that a constant Fermi level is used along each individual slice in the Y direction for carrier concentration calculation in the Schrodinger - Poisson model.

HANSCHQM turns on the Hansch quantum effects approximation model for N channel MOS devices.

POST.SCHRO specifies to calculate the Schrodinger equation for quantum effects as a post processing step.

SCHRO enables the Poisson-Schrodinger solver. This can be used for zero carrier solutions only (specified by METHOD CARRIERS=0). This is typically combined with the EIGENS parameter to control the number of eigenstates calculated. This is a 1D solver that is solved within the mesh between limits set by QX.MIN and QX.MAX.

N.DORT turns on the Van Dort quantum effects approximation model for N channel MOS devices.

P.DORT turns on the Van Dort quantum effects approximation model for P channel MOS devices.

QUANTUM enables the quantum moments model.

QX.MIN, QX.MAX specify the minimum and maximum extent of the Poisson-Schrodinger solver along the x-axis direction.

Energy Balance Simulation Flags

E.TAUR.VAR specifies that electron temperature dependent energy relaxation time is used. Use parameters TRE.T1, TRE.T2, TRE.T3, TRE.W1, TRE.W2 and TRE.W3 on material statement to specifie the energy relaxation time.

H.TAUR.VAR specifies that hole temperature dependent energy relaxation time is used. Use parameters TRH.T1, TRH.T2, TRH.T3, TRH.W1, TRH.W2 and TRH.W3 on material statement to specifie the energy relaxation time.

HCTE specifies that both electron and hole temperature will be solved.

HCTE.**EL** specifies that electron temperature will be solved.

HCTE.**HO** specifies that hole temperature will be solved.

F.KSN specifies the name of a file containing a C-INTERPRETER function specifying the electron Peltier coefficient as a function of electron energy.

F.KSP specifies the name of a file containing a C-INTERPRETER function specifying the hole Peltier coefficient as a function of hole energy.

KSN specifies which hot carrier transport model will be used for electrons. KSN=0 selects the hydrodynamic model and KSN=-1 selects the energy balance model.

KSP specifies which hot carrier transport model will be used for holes. KSP=0 selects the hydrodynamic model and KSP=-1 selects the energy balance model.

TAUMOB specifies the dependence of relaxation times with carrier temperature in the mobility definition. If TAUMOB is specified, the values of MATERIAL statement parameters TAUMOB.EL and TAUMOB.HO are dependent on the carrier temperature.

TAUTEM specifies the dependence of relaxation times with carrier temperature. If TAUTEM is specified, the values of MATERIAL statement parameters TAUREL.EL and TAUREL.HO are dependent on carrier temperature..

N.**TEMP** or **HCTE.EL** specifies that the electron temperature equation will be solved.

P.**TEMP** or **HCTE.HO** specifies that the hole temperature equation will be solved.

Lattice Heating Simulation Flags

Note: These parameters should only be specified if GIGA or GIGA3D is enabled on your system.

HEAT.FULL enables all thermal sources and sinks (Joule heat, generation-recombination heat, and Peltier Thomson heat).

HEAT.PETHOM can be used to turn off the Peltier-Thomson heat source in the HEAT.FULL option.

LAT.TEMP or **L.TEMP** specifies that the lattice temperature equation will be solved. For lattice heating simulation there must be at least one thermal contact defined using the THERMCONTACT statement.

Model Macros

BIPOLAR selects a default set of models which are used when simulating bipolar devices. The bipolar models are CONMOB, FLDMOB, BGN, CONSRH, and AUGER.

ERASE specifies a default set of models which are used to simulate EEPROM erasure. When is specified: the MOS, FNORD, IMPACT, and BBT.KL models will be used.

MOS specifies a default set of models for MOS devices. The MOS models are CVT, SRH and FERMI.

PROGRAM specifies a default set of models used when writing to EEPROMS. When PROGRAM is specified: the MOS, HEI, and IMPACT models will be used.

General Parameters

DEVICE specifies which device in MIXEDMODE simulation that the MODELS statement should apply to.

DRIFT.DIFF specifies that the drift-diffusion transport model is to be used. This implies that the electron and hole carrier temperature equations will not be solved.

MATERIAL specifies which material from the table in Appendix B the MODELS statement should apply to. If a material is specified then all regions defined as being composed of that material will be affected.

NAME specifies in which region the MODELS statement should apply. Note that the name must match the name specified in the NAME parameter of the REGION statement.

PRINT prints the status of all models, a variety of coefficients, and constants. **I**t is recommended to include this parameter in all ATLAS runs.

SI20 sets a compatibility mode for the materials Si and SiGe with ATLAS versions prior to version 3.0.

TEMPERATURE specifies the temperature in Kelvin.

Model Dependent Parameters

ARORA Model.

Parameters which may be used only with the ARORA model (see Equations 3-126 and 3-127) include AR.MU1N, AR.MU2N, AR.MU1P, and AR.MU2P.

BBT.KL and BBT.STD Models.

Parameter used with the BBTKL model are BB.A1 and BB.B. Parameters used with the BBT.STD model are BB.A2 and BB.B. Parameters BB.A1 and BB.A2 are the pre-exponential coefficients in the band-to-band tunneling models. The parameter BB.B is the exponential coefficient used in both models. The default value of BB.B depends on which model is chosen (see Equation 3-270).

CCSMOB Model.

The parameters CCS.EA, CCS.EB, CCS.HA, and CCS.HB describe the dependendence of mobility on doping, carrier concentration, and temperature. See Equations 3-128 – 3-131.

FLDMOB Model.

B.**ELECTRONS** is used in the field-dependent mobility expression for EVSATMOD=0 (see Equation 3-198).

B.**HOLES** is used in the field-dependent mobility expression for EVSATMOD=0 (see Equation 3-199).

E0 is used in the field dependent mobility model for EVSATMOD=1 (see Equation 5-50).

EVSATMOD specifies which parallel field dependent mobility model (see Equations 3-198 and 5-50) should be used for electrons as follows:

0. Use the standard saturation model

1. Use the negative differtial velocity saturation model

2 Use a simple velocity limiting model.

In most cases the default value of 0 should be used.

HVSATMOD specifies which parallel field dependent mobility model (see Equations 3-199 and 5-51) should be used for holes as follows:

0. Use the standard saturation model

2 Use a simple velocity limiting model.

In most cases the default value of 0 should be used.

Fowler-Nordheim Tunneling Model.

Parameters used in this model are F.AE and F.BE. F.AE is the pre-exponential factor and F.BE is the exponenential coefficient (see Equation 3-274).

WATT or SURFMOB Model.

The parameters which may be used with the WATT model include ALN1, ALN2, ALN3, ALP1, ALP2, ALP3, ETAN, ETAP, MREFN1, MREFN2, MREFN3, MREFP1, MREFP2, and MREFP3 (see Equations 3-190 and 3-191).

TFLDMB1 and TFLDMB2 Models.

ACC.**SF** specifies the accumulation saturation factor which describes the ratio of the majority carrier concentration in the accumulation layer before and after bending of conductivity and valence bands.

INV.**SF** specifies the inversion saturation factor which describes the ratio of the majority carrier concentration in the inversion layer before and after the bending of conductivity and valance bands.

OX.**BOTTOM** specifies the coordinate of the bottom edge of the gate oxide for a MOSFET transistor.

OX.**LEFT** specifies the coordinate of the left edge of the gate oxide for a MOSFET transistor.

OX.**RIGHT** specifies the coordinate of the right edge of the gate oxide for a MOSFET transistor.

CONCANNON Model Parameters

CGATE.**N** specifies an empirical tuning factor for electrons in the Concannon gate current model.

CGATE.**P** specifies an empirical tuning factor for holes in the Concannon gate current model.

PEFF.**N** specifies an effective barrier height for electrons in the Concannon gate current model.

PEFF.**P** specifies an effective barrier height for holes in the Concannon gate current model.

THETA.**N** specifies the critical rejection angle for electrons in the Concannon gate current model.

THETA.**P** specifies the critical rejection angle for electrons in the Concannon gate current model.

C0 specifies the electron distribution weight factor in the Concannon gate current model.

CHIA specifies the electron distribution function constant in the Concannon gate current model.

CHIB specifies the electron distribution function constant in the Concannon gate current model.

CHI.**HOLES** specifies the hole distribution function constant in the Concannon gate current model.

ENERGY.**STEP** specifies the energy step for numeric integration in the Concannon gate current model.

INFINITY specifies the highest energy in numeric integration in the Concannon gate current model.

PATH.**N** specifies the mean free path in the oxide for electrons in the Concannon gate current model.

PATH.**P** specifies the mean free path in the oxide for holes in the Concannon gate current model.

HEI Model Parameters

The parameters which may be used with the `HEI` model include `IG.ELINR`, `IG.HLINR`, `IG.ELINF`, `IG.HLINF`, `IG.EBETA`, `IG.HBETA`, `IG.EETA`, `IG.HETA`, `IG.EEOX`, `IG.HEOX`, `IG.EB0`, `IG.HB0`.

N.DORT and P.DORT Model parameters

B.DORT user specifiable model parameter for the van Dort quantum effects approximation model.

D.DORT user specifiable model parameter for the van Dort quantum effects approximation model.

LASER Simulation Parameters

CAVITY.**LENGTH** specifies the cavity length in the longitudinal direction (in μm).

GAINMOD specifies the local optical gain model to be used.

- GAINMOD=0 specifies that no optical gain model will be used.
- GAINMOD=1 specifies that the complex frequency-dependent gain model will be used (see Equation 9-3).
- GAINMOD=2 specifies that the simple gain model will be used (see Equation 9-6).

LASER enables LASER simulation.

LMODE specifies that multiple longitudinal models are to be taken into account during LASERT simulation.

LAS.ABSORPTION enables the absorption loss model in LASER.

LAS.**EINIT**, **LAS**.**EFINAL** specify the lower and upper photon energies. LASER will calculate multiple longitudinal photon rates within this range. Using wide ranges can slow down simulation.

LAS.ESEP specifies the photon energy separation. If this is not specified, LASER will automatically calculate the number of longitudinal modes based on the cavity length and the energy range.

LAS.FCARRIER enables the free carrier loss model in LASER.

LAS.**ITMAX** specifies the maximum number of iterations allowed for LASER simulation at each bias point.

LAS.**LOSSES** specifies the total losses in Equation 9-8.

LAS.**MAXCH** specifies the maximum allowable relative charge in photon densities between iterations. Rapid changes of the photon densities can cause convergence problems.

LAS.MIRROR specifies the percentage facet reflectivity (both facets are assumed to have this value of reflectivity) for the mirror loss in Laser. 100 % reflectivity is equalivent to no mirror loss.

LAS.MULTISAVE specifies the whether or not to save the transient LASER spectrum as one file or multiple files.

LAS.**NEFF** specifies the effective refractive index in Equation 9-2.

LAS.**OMEGA** specifies the lasing frequency to be used in Equation 9-1. If model 2 is used for simulation, then this parameter specifies an estimate of the lasting frequency. Instead of using this parameter, PHOTON.ENERGY can be used to specify photon energy.

LAS.**SIN** specifies an initial photon density in the fundamental lasing mode. This value provides an initial guess for subsequent iterations. This parameter is used only when the single frequency model has been selected.

LAS.SPECSAVE the spectrum file will be saved after every LAS.SPECSAVE LASER solution steps.

LAS.**TAUSS** specifies the iteration parameter to be used for the photon rate equation when solving Equation 9-8.

LAS.TIMERATE specifies that the time dependent photon rate equation will be used in a transient laser simulation.

LAS.**TOLER** specifies the desired accuracy in photon areas.

PHOTON.**ENERGY** specifies the energy of photons. If model 2 is used for simulation, this parameter specifies only an initial estimate of the photon energy. Instead of using this parameter, LAS.OMEGA can be used to specify the lasing frequency.

SPEC.**NAME** specifies the name of a spectrum file, which LASER will produce for each bias point if the LMODES parameter has been specified.

Model Selection Example

This example selects concentration and field dependent mobilities, SRH recombination, and Auger recombination. This is a typical model set for bipolar simulation. The simulation temperature is 290K.

```
MODELS CONMOB FLDMOB SRH AUGER TEMP=290
```

Confirming Model Selection

To echo back model selections, parameters and material constants use

```
MODELS PRINT
```

Note: *For the best MODEL selection for different applications, consult the Standard Example set.*

OPTIONS

OPTIONS — sets options for an entire run.

Syntax

```
OPTIONS  [<rcp>]
```

Parameter	Type	Default	Units
NORMAL	Logical	True	
QUIET	Logical	False	
VERBOSE	Logical	False	

Description

rcp is one or more of the run control parameters described below. These parameters, which are not normally used, specify debugging options.

NORMAL is the default specification for run-time output filtering. At this setting ATLAS prints out the most relevant information (e.g. mesh statistics, terminal voltages, currents, warnings and error messages).

QUIET specifies the maximum of filtering of run-time output.

VERBOSE specifies the mimimum filtering of run-time output. You should specify VERBOSE if you want output of residual norms.

Example

The following syntax can be used to provide extra debugging output.

```
OPTION  VERBOSE
```

Note: The VERBOSE output is of little interest to most users

OUTPUT

OUTPUT — allows the user to specify the data that will be stored in standard structure format files.

Syntax

```
OUTPUT <parameters>
```

Parameter	Type	Default	Units
ANGLE	Logical	False	
AREA	Logical	False	
BAND.PARAM	Logical	False	
CHARGE	Logical	False	
CON.BAND	Logical	False	
CONTACT	Integer		
DELTAV	Real	0.1	Volts
DEVDEG	Logical	False	
E.FIELD	Logical	False	
E.LINES	Logical	False	
E.MOBILITY	Logical	False	
E.TEMP	Logical	True	
E.VELOCITY	Logical	False	
EFIELD	Logical	False	
EIGENS	Integer	7	
EX.FIELD	Logical	True	
EX.VELOCITY	Logical	False	
EY.FIELD	Logical	True	
EY.VELOCITY	Logical	False	
FLOWLINES	Logical	False	
H.MOBILITY	Logical	False	
H.TEMP	Logical	True	
H.VELOCITY	Logical	False	
HX.VELOCITY	Logical	False	
HY.VELOCITY	Logical	False	
IMPACT	Logical	True	
INV.AREA	Logical	False	
INV.ANGLE	Logical	True	

Parameter	Type	Default	Units
INAME	Character		
JX.ELECTRON	Logical	False	
J.ELECTRON	Logical	True	
JX.ELECTRON	Logical	False	
J.CONDUC	Logical	True	
J.DISP	Logical	False	
J.HOLE	Logical	True	
J.TOTAL	Logical	True	
JX.CONDUC	Logical	False	
JX.HOLE	Logical	False	
JX.TOTAL	Logical	False	
JY.CONDUC	Logical	False	
JY.HOLE	Logical	False	
JY.TOTAL	Logical	False	
KSN	Logical	False	
KSP	Logical	False	
LRATIO	Real	1.0	
MINSET	Logical	False	
N.LINES	Integer		
OPT.INTENS	Logical	False	
OX.CHARGE	Logical	False	
OLD.AVG	Logical	False	
PERMITTIVITY	Logical	False	
PHOTOGEN	Logical	True	
QFN	Logical	True	
QFP	Logical	True	
QSS	Logical	False	
RECOMB	Logical	True	
TAURN	Logical	False	
TAURP	Logical	False	
TOT.DOPING	Logical	False	
TRAPS	Logical	True	

Parameter	Type	Default	Units
U.AUGER	Logical	False	
U.RADIATIVE	Logical	False	
U.SRH	Logical	False	
VAL.BAND	Logical	False	
VECTORS	Logical	False	
X.COMP	Logical	True	
Y.COMP	Logical	False	

Description

BAND.**PARAM** specifies that the band parameters (Eg, ni, Ne, Nv, and c) are included in the standard structure file.

CHARGE specifies that the net charge will be included in the standard structure file.

CON.**BAND** specifies that the conduction band edge will be included in the standard structure file.

DEVDEG causes the distribution of acceptor/donor like traps on the interface, hot electron/hole current density on the interface, and trapped electron/holes to be written to the structure file.

E.**FIELD** or **EFIELD** specifies that total electric field will be included in the standard structure file.

E.**LINES** specifies the elecrtric field lines will be included in the standard structure file.

E.**MOBILITY** specifies that electron mobility will be included in the standard structure file.

E.**TEMP** specifies that the electron temperature will be included in the standard structure file.

E.**VELOCITY** specifies that the total electron velocity will be included in the standard structure file.

EIGENS specifies the maximum number of eigen energies and eigen functions to be written to the structure file from a Poisson- Schrodinger solution.

EX.**FIELD** specifies that the x-component of electric field will be included in the standard structure file.

EX.**VELOCITY** specifies that the x-component of electron velocity will be included in the standard structure file.

EY.**FIELD** specifies that the y-component of electric field will be included in the standard structure file.

EY.**VELOCITY** specifies that the y-component of electron velocity will be included in the standard structure file.

FLOWLINES specifies that the current flowlines will be included in the standard structure file.

H.**MOBILITY** specifies that hole mobility will be included in the standard structure file.

H.**TEMP** specifies that the hole temperature will be included in the standard structure file.

H.**VELOCITY** specifies that the total hole velocity will be included in the standard structure file.

HX.**VELOCITY** specifies that the x-component of hole velocity will be included in the standard structure file.

HY.**VELOCITY** specifies that the y-component of hole velocity will be included in the standard structure file.

IMPACT specifies that the impact ionization rate will be included in the standard structure file.

J.**CONDUC** specifies that the total conduction current density will be included in the standard structure file.

J.**DISP** specifies that the total displacement current density will be included in the standard structure file.

J.**ELECTRON** specifies that the total electron current density will be included in the standard structure file.

J.**HOLE** specifies that the total hole current density will be included in the standard structure file.

J.**TOTAL** specifies that the total current density will be included in the standard structure file.

JX.**CONDUC** specifies that the x-component of the total conduction current density will be included in the standard structure file.

JX.**ELECTRON** specifies that the x-component of electron current density will be included in the standard structure file.

JX.**HOLE** specifies that the x-component of hole current density will be included in the standard structure file.

JX.**TOTAL** specifies that the x-component of total current density will be included in the standard structure file.

JY.**CONDUC** specifies that the y-component of the total conduction current density will be included in the standard structure file.

JY.**ELECTRON** specifies that the y-component of electron current density will be included in the standard structure file.

JY.**HOLE** specifies that the y-component of hole current density will be included in the standard structure file.

JY.**TOTAL** specifies that the y-component of total current density will be included in the standard structure file.

KSN specifies that electron Peltier coefficients are to be written to any saved structure file.

KSP specifies that hole Peltier coefficients are to be written to any saved structure file.

MINSET that a minimum set of data (potential, carrier concentration, and electric field) will be included in the standard structure file.

OPT.**INTENS** specifies that optical intensity is included in the standard structure file.

OX.**CHARGE** specifies that fixed oxide charge is inclued in the standard structure file.

PERMITTIVITY specifies the dielectric permittivity is saved

PHOTOGEN specifies that the photogeneration rate will be included in the standard structure file.

QFN specifies that the electron quasi-fermi level will be included in the standard structure file.

QFP specifies that the hole quasi-fermi level will be included in the standard structure file.

QSS specifies that the surface charge will be included in the standard structure file.

RECOMB specifies that the recombination rate will be included in the standard structure file.

TAURN specifies that electron relaxation times are to be written to any saved structure file.

TAURP specifies that hole relaxation times are to be written to any saved structure file.

TOT.**DOPING** specifies that total doping will be included in the standard structure file.

TRAPS specifies that trap density information will be included in the standard structure file.

U.**AUGER** specifies that the Auger component of recombination is to be written to solution files.

U.**RADIATIVE** specifies that the radiative component of recombination is to be written to solution files.

U.**SRH** specifies that the SRH component of recombination is to be written to solution files.

VAL.**BAND** specifies that the valence band edge will be included in the standard structure file.

VECTORS specifies that only vector components will be included in the standard structure file.

X.**COMP** specifies that the composition fraction “x” is to be written to solution files.

Y.**COMP** specifies that the composition fraction “y” is to be written to solution files.

Ionization Integral Parameters

Note: See the SOLVE statement and the on-line examples for instructions on using ionization integrals

INAME specifies the name of a contact for which electric field lines are calculated.

CONTACT specifies a contact number for which electric field lines are calculated.

LRATIO specifies the spacing ratio between adjacent electric field lines. Defaults to 1.0 for uniform spacing.

N.LINES specifies the number of electric field lines.

DELTAV since the electric field is near zero at the contact, the electric field line calculations begin at a distance from the contact at which the contact voltage has changed by DELTAV. Defaults to 0.1 V.

Averaging Parameters for Vector Quantities

OLD.AVG specifies that the current and field quantities will be averaged using an older algorithm (from version 3.0.0.R and back). By default the new method is used.

ANGLE specifies that averaging of current and fields will be weighted by the size of the angle of triangles intersecting at the node.

INV.ANGLE specifies that averaging of current and fields will be weighted by the inverse of the size of the angle of triangles intersecting at the node.

AREA specifies that averaging of current and fields will be weighted by the areas of triangles intersecting at the node.

INV.AREA specifies that averaging of current and fields will be weighted by the inverse of areas of triangles intersecting at the node.

Note: Certain quantities that can be output into the structure file and subsequently displayed using TONYPLOT need special mention. These quantities are evaluated within ATLAS along the links between grid points. They are represented in the structure file at the grid points themselves. As such these quantities are subject to averaging. In particular, electric field and currents are averaged so as to take into account the vector nature of these values. Mobility is simply summed up over all the links surrounding the grid point and divided by the total number of links. Carrier velocities are derived by dividing the averaged current by the carrier density at the grid point and the fundamental electron charge q.

Example of combining OUTPUT with SOLVE and SAVE

The OUTPUT statement is often used in conjuction with the SAVE statement. The following statement lines specify that current flowlines and electron velocity components are saved in all subsequent standard structure solution files.

```
OUTPUT FLOWLINES EX.VELO EY.VELO
SOLVE  PREVIOUS V5=2 OUTF=data1.str MASTER
SAVE  OUTF=data2.str
```

PROBE

PROBE — allows users to output the minimum value, maximum value or value at a specified location of any of several distributed quantities to the LOG file. The quantities will be saved at each bias point of timestep in the LOG file.

Note: PROBE is the most accurate way to determine the value of many parameters calculated by ATLAS. Parameters stored on node points in the structure files for Tonyplot are often interpolated and subject to noise.

Syntax

```
PROBE[MIN|MAX|X=<n>Y=<n>[DIR=<n>]][POTENTIAL|N.CONC|P.CONC|N.TEMP|
P.TEMP|LAT.TEMP|GENERATION|RECOMBINATION|RSRH|RAUGER|RRADIATIVE]
```

Parameter	Type	Default	Units
DEVICE	Character		
DIR	Real	0.0	degrees
FIELD	Logical	False	
LAT.TEMP	Logical	False	
MAX	Logical	False	
MIN	Logical	False	
N.CONC	Logical	False	
N.MOB	Logical	False	
N.TEMP	Logical	False	
NAME	Character		
P.CONC	Logical	False	
P.MOB	Logical	False	
P.TEMP	Logical	False	
POLARIZATION	Logical	False	
POTENTIAL	Logical	False	
QFN	Logical	False	
QFP	Logical	False	
RAUGER	Logical	False	
RECOMBIN	Logical	False	
RRADIATIVE	Logical	False	

Parameter	Type	Default	Units
RSRH	Logical	False	
X	Real		μm
Y	Real		μm

Description

DEVICE specifies which device in mixed mode simulation the probe statement should apply to.

DIR specifies the direction relative to the **x** axis in degrees associated with certain directed quantities. These quantities include FIELD, N.MOB, P.MOB and POLARIZATION.

Note: The algorithm used finds the triangle in the mesh containing the specified X and Y values. Then the value of the DIR parameter is used to find which edge of the triangle lies in the direction nearest that value

---

FIELD - specifies that a value of electric field is probed. The DIR parameter should also be specified if FIELD is used.

GENERATION specifies that the generation rate due to impact ionization is probed.

LAT.TEMP specifies that the probe will operate on lattice temperature.

MAX specifies that the probe will find the maximum value on the mesh.

MIN specifies that the probe will find the minimum value on the mesh.

NAME sets a character string that allows the user to specify the description displayed by TONYPLOT.

N.CONC specifies that the probe will operate on electron concentration.

N.MOB specifies that the probe will operate on the electron mobility. The DIR parameter should also be specified if N.MOB is used.

N.TEMP specifies that the probe will operate on electron temperature.

P.CONC specifies that the probe will operate on hole concentration.

P.MOB specifies that the probe will operate on the hole mobility. The DIR parameters should also be specified if P.MOB is used.

P.TEMP specifies that the probe will operate on hole temperature.

POLARIZATION specifies that the probe will operate on ferroelectric polarization.The DIR parameter should also be specified if POLARIZATION is used.

POTENTIAL specifies that the probe will operate on electrostatic potential.

QFN specifies that the probe will operate on the electron quasi-Fermi level.

QFP specifies that the probe will operate on the hole quasi-Fermi level.

RAUGER specifies that the probe will operate on Auger recombination rate.

RECOMBIN specifies that the probe will operate on net recombination rate.

RRADIATIVE specifies that the probe will operate on radiative recombination rate.

RSRH specifies that the probe will operate on SRH recombination rate.

Example of Probing the Maximum Value

The following line will cause the maximum electron concentration on the grid to be output to the log file:

```
PROBE NAME=peak_electrons MAX N.CONC
```

Example of PROBE at a location

This syntax will cause the potential at the location X=0.5 Y=0.1 to be output to the log file:

```
PROBE NAME=mypotential X=0.5 Y=0.1 POTENTIAL
```

Vector Quantitiy Example

For vector quantities the direction parameter `DIR` must be specified.These two lines allow a lateral mobility and vertical field in a MOSFET.

```
PROBE NAME=channel_mobility x=1 y=0.001 N.MOB DIR=0
PROBE NAME=channel_field x=1 y=0.001 FIELD DIR=90
```

QUIT

QUIT — stops execution of ATLAS.

Syntax

```
QUIT
```

Synonyms

END

EXIT

Description

The QUIT statement may be placed anywhere in an input file. ATLAS will stop execution upon encountering the QUIT statement. All input lines after the occurrence of the QUIT statement will be ignored for that execution of ATLAS.

Note: To quit and immediately restart ATLAS inside of DECKBUILD the GO ATLAS statement should be used. Full details on the GO syntax are in the VWF INTERACTIVE TOOLS MANUAL VOLUME ONE

REGION

REGION — specifies the location of materials in a previously defined mesh. Every triangle must be defined as a material.

Syntax

```
REGION NUMBER=<n> <material> [<position>]
```

Parameter	Type	Default	Units
GRAD.12	Real		μm
GRAD.23	Real		mm
GRAD.34	Real		μm
GRAD.41	Real		μm
IX.LOW	Integer	left of structure	
IX.HIGH	Integer	right of structure	
IY.LOW	Integer	top of structure	
IY.HIGH	Integer	bottom of structure	
IZ.HIGH	Real		
IZ.LOW	Real		
MATERIAL	Character		
NAME	Character		
NUMBER	Integer		
X.COMP	Real	0.0	
Y.COMP	Real	0.0	
X.MIN	Real	left of structure	μm
X.MAX	Real	right of structure	μm
Y.MIN	Real	top of structure	μm
Y.MAX	Real	bottom of structure	μm
Z.MAX	Real		
Z.MIN	Real		

Description

n specifies a region number from 1 to 200.

material is one or more of the material names described below.

position is one or more of the position parameters described below.

Material Parameters

MATERIAL specifies the material used for the region. Valid material names are listed in the table in Appendix B.All materials are divided into three classes: semiconductors, insulators and conductors. See Appendix B for important information about requirements for each material class.

NAME specifies the name of the region. The name can be used in the MODELS, MATERIAL, and IMPACT statements to provide regionally dependent models.This name is just a label any does not imply any material parameter settings.

NUMBER assigns a region number. Multplie REGION lines with the same number can be used to define region shapes made from several rectangles.

Note: If REGION definitions overlap the highest region number takes precedence

---

GRAD.<n> specifies the compositional gradings for heterojunctions along each side of the region rectangle or quadrillateral. The value of the GRAD parameters specifies the distance at which the composition fraction reduces to zero. A value of 0.0 specifies that the heterojunction is abrupt. The value of <n> can be the numbers 12, 23, 34, and 41. These correspond to the point indices around the rectangular region working clockwise from top left.

- 12 = top surface
- 23= right hand side
- 34= bottom surface
- 41 = left hand side

Position Parameters

You may use grid indices to define a region only when the mesh is rectangular. To define a region with a rectangular mesh:

1. The X.MESH and Y.MESH statements must be used to specify grid indices.
2. The IX.HIGH, IX.LOW, IY.HIGH, and IY.LOW parameters may be used to specify x and y mesh line number values.

Note: To add regions to irregular meshes, such as those read from ATHENA, boundaries must be specified using the X.MAX, X.MIN, Y.MAX, and Y.MIN parameters.

IX.HIGH specifies the maximum x value of the grid index.

IX.LOW Specifies the minimum x value of the grid index.

IY.HIGH specifies the maximum y value of the grid index.

IY.LOW specifies the minimum y value of the grid index.

IZ.HIGH specifies the maximum z value of the grid index.

IZ.LOW specifies the mimimum z value of the grid index.

X.COMP is the composition fraction "X" for a region with a composition dependent cations (e.g. AlGaAs).

X.MAX specifies the maximum x-boundary.

X.MIN specifies the minimum x-boundary.

Y.COMP is the composition fraction "Y" for a region with a composition dependent anions(e.g. InGaAsP).

Y.MAX specifies the maximum y-boundary.

Y.MIN specifies the minimum x-boundary.

Z.MIN specifies the minimum z-boundary.

Z.MAX specifies the maximum z-boundary.

Grid Inidices Example

Define a silicon region extending from nodes 1 to 25 in the x-direction and from nodes 1 to 20 in the y-direction.

```
REGION NUM=1 IX.LO=1 IX.HI=25 IY.LO= 1 IY.HI=20 MATERIAL=SILICON
```

Non-Rectangular Region Example

Define a region which is composed of two separate retangular areas. Note that region statements are cumulative in effect.

```
REGION NUM=1 IX.LO=4 IX.HI=5 IY.LO=1 IY.HI=20 MATERIAL=OXIDE
REGION NUM=1 IX.LO=1 IX.HI=30 IY.LO=1 IY.HI=37 MATERIAL=OXIDE
```

Typical MOS Example

Define regions for a typical MOS structure.

```
REGION NUM=1 Y.MAX=0 MATERIAL=OXIDE
REGION NUM=2 Y.MIN=0 MATERIAL=SILICON
```

3D Region Definition Example

Define a cube of oxide within a region silicon in 3D.

```
REGION NUM=1 MATERIAL=SILICON
REGION NUM=2 Y.MAX=0.5 X.MIN=0.5 \
   X.MAX=1.0 Z.MIN=0.5 Z.MAX=1.0 MATERIAL = OXIDE
```

Graded Heterojunction Defintion Example

Define a graded heterojunction of AlGaAs/GaAs

```
REGION NUM=1 MATERIAL=GaAs Y.MIN=1
REGION NUM=2 MATERIAL=AlGaAs Y.MAX=0.9 X.COMP=0.2 GRAD.34=0.1
```

In this case the area between y=0.9 and 1.0 is graded in composition from 0.2 to 0.0. The parameter `Y.MAX` refers to the bottom of the uniform composition region. The actual bottom of the AlGaAs region is `Y.MAX+GRAD.34`

REGRID

REGRID — allows the user to refine a crude mesh. A triangle is refined when the chosen variable changes by more than a specified criteria.

Syntax

```
REGRID RATIO=<n> <var> [<lp>] [<cp>] [<io>]
```

Parameter	Type	Default	Units
ABSOLUTE	Logical	False	
ASCII	Logical	False	
CHANGE	Logical	True	
COS.ANG	Real	2.0	
DOPFILE	Character		
DOPING	Logical		cm^{-3}
E.TEMP	Logical	False	
EL.FIELD	Logical		V/cm
ELECTRON	Logical		cm^{-3}
H.TEMP	Logical	False	
HOLE	Logical		cm^{-3}
IGNORE	Integer		
IN.GREEN	Character		
LOCALDOP	Logical	False	
LOGARITHM	Logical	True	
MAX.LEVEL	Integer	1+ maximum level of grid	
MIN.CARR	Logical		cm^{-3}
NET.CARR	Logical		cm^{-3}
NET.CHRG	Logical		cm^{-3}
PISCES.OUT	Logical	False	
OUT.GREEN	Character	value of OUTFILE + tt	
OUTFILE	Character		
POTENTIAL	Logical		V
QFN	Logical		V
QFP	Logical		V

Parameter	Type	Default	Units
REGION	Integer		All
SMOOTH.K	Integer		
STEP	Real		
X.MAX	Real	Right	µm
X.MIN	Real	Left	µm
Y.MAX	Real	Bottom	µm
Y.MIN	Real	Top	µm

Description

RATIO or **STEP** — specifies the maximum allowed variance across one element.

var is one of the variable parameters described below. The selected parameter is used as the basis for regridding.

lp is one of more of the location parameters described below. These parameters are used to select the areas which are to be refined.

cp is one or more of the control parameters. These parameters are used to control the plotted output.

io is one or more of the File I/O parameters .

Variable Parameters

DOPING selects net doping.

E.**TEMP** select electron temperature.

EL.**FIELD** selects electric field.

ELECTRON selects electron concentration.

H.**TEMP** selects hole temperature.

HOLE selects hole concentration.

MIN.**CARR** selects minority carrier concentration.

NET.**CARR** selects net carrier concentration.

NET.**CHRG** selects net charge.

POTENTIAL selects mid-gap potential.

QFN selects the electron quasi-Fermi level.

QFP selects the hole quasi-Fermi level.

Location Parameters

If no location parameters are specified, refinement will include:

1. All regions for potential and electric field regrids
2. All semiconductor regions for regrids which depend on the other variables

IGNORE specifies regions that are not to be refined.

REGION specifies regions which are refined according to the specified critierion. Other regions may be refined to maintain well-shaped triangles.

X.MAX uses device coordinates to specify the maximum x-value for refinement.

X.MIN uses device coordinates to specify the minimum x-value for refinement.

Y.MAX uses device coordinates to specify the maximum y-value for refinement.

Y.MIN uses device coordinates to specify the minimum y-value for refinement.

Control Parameters

ABSOLUTE specifies that the absolute value of the variable be used.

CHANGE determines whether to use the magnitude or the difference of a triangle variable as the refinement criterion. This parameter defaults to "difference".

COS.ANGLE limits the creation of obtuse angles in the mesh by specifying "obtuse criterion". If this parameter is used, nodes are added to the mesh so that the number of obtuse triangles is reduced.

Note: Be careful when using the COS.ANGLE parameter. Recommended values are from 0.8 to 0.95. Smaller values may dramatically increase the number of nodes.

LOCALDOP specifies that if minority carrier concentration exceeds local doping, the grid will be refined. This parameter is used in conjunction with minority carrier regrids.

LOGARITHM specifies a logarithmic refinement scale. Since many of the quantities may become negative, numerical problems are avoided by using $\log(x)=\text{sign}(x)\cdot\log10(1+|x|)$. If you wish to obtain the true logarithm of a quantity, the ABSOLUTE parameter must be specified before the LOGARITHM parameter is specified. The absolute value of a quantity is computed first, thereby eliminating negative arguments.

MAX.LEVEL specifies the maximum level of any triangle relative to the original mesh. This parameter defaults to one more than the maximum level of the grid, but can be set to a smaller value to limit refinement. Values less than or equal to zero are interpreted relative to the current maximum grid level.

SMOOTH.KEY specifies a smoothing index. The digits of the index are read in reverse order and interpreted as follows:

1. Triangle smoothing. All region boundaries remain fixed.
2. Triangle smoothing. Only material boundaries are maintained.
3. Node averaging.
4. Improved triangle smoothing method. This method uses diagonal flipping to reduce the number of obtuse triangles.
5. Aligns triangles with electric field gradient.

Usually option 1 is sufficient. Option 2 is useful only if a device has several regions of the same material and the border between different regions is unimportant. Option 3 is not recommended when the initial mesh is basically rectangular, such as mesh information usually obtained from SSUPREM4. Option 4 is similar to option 1, but option 4 usually creates less obtuse triangles.

File I/O Parameters

ASCII specifies that mesh files and triangle trees will be written in an ASCII rather than a binary format. This parameter has no effect on the device doping file (see the DOPFILE parameter).

DOPFILE specifies the name of a file which contains device doping information. This file is created on the DOPING statement. Specifying DOPFILE avoids linear interpolation of doping values at newly created grid points by using the initial doping specification to apply doping to the new grid points.

IN.GREEN specifies a triangle tree for the mesh which will be used in this regrid. If this parameter is not specified, the program will look for a file with the same name as the current mesh plus *tt* at the end. If no such file exists, the program will not use a triangle tree for the previous mesh.

MASTER.OUT saves mesh and doping information in a standard structure file format.

OUTFILE specifies the name of a standard structure output file where mesh information will be stored. This parameter must be specified if the mesh is to be used for subsequent runs.

OUT.GREEN specifies the name of the file that holds the history of the triangle tree. This history is used in further regrid steps.

PISCES.OUT saves mesh and doping information in a binary PISCES-II format.Files in this format cannot be displayed in TONYPLOT

Doping Regrid Examples

Starting with an initial grid, we refine twice so that all triangles with large doping steps are refined.

```
REGRID LOG DOPING RATIO=6 OUTF=grid1 DOPF=dopxx SMOOTH=4
REGRID LOG DOPING RATIO=6 OUTF=grid2 DOPF=dopxx SMOOTH=4
```

A similar effect could be obtained with just one regrid statement.

```
REGRID LOG DOPING RATIO=6 OUTF=grid2 DOPF=dopxx MAX.LEVEL=2
```

In both cases, two levels of refinement are performed. The first example is recommended because new doping information is introduced at each level of refinement. This produces better refinement criterion and fewer triangles.

Potential Regrid Example

Next, an initial solution is produced and triangeles which exhibit large potential steps are refined.

```
SOLVE INIT
REGRID POTENTIAL RATIO=0.2 OUTF=grid3.str SMOOTH=4
```

Re-initializing after regrid example

Often it is necessary to re-solve the same bias point after a REGRID using the following style of syntax.

```
SOLVE VDRAIN=3.0
REGRID  POTENTIAL RATIO=0.25 SMOOTH=4 OUTF=mygrid.str
SOLVE PREV
```

Occasionally it is necessary to quit and restart ATLAS with the new mesh. This is done using syntax such as:

```
SOLVE VDRAIN=3.0
REGRID  POTENTIAL RATIO=0.25 SMOOTH=4 OUTF=mygrid.str
go atlas
MESH INF=mygrid.str
```

After this MESH statement all models, material parameters and numerical methods have to be respecified before any SOLVE statement.

SET

The SET statement is used to define variables for substitution into ATLAS syntax. SET commands are executed by DECKBUILD.

Note: Full documentation of the SET statement is in the VWF INTERACTIVE TOOLS MANUAL VOLUME ONE

Numeric Variable Example

Define a numerical variable. Use it in a calculation and substitute it into the ATLAS syntax for REGION definition

```
SET MYLENGTH=0.1
SET HALFLENGTH= $"MYLENGTH"*0.5
...
REGION NUM=1 MATERIAL=SILICON X.MIN=$"HALFLENGTH" X.MAX=$"MYLENGTH"
```

String Variable Example

Define a string variable to use as part of a filename

```
SET LABEL=testcase1
..
LOG OUTF=$"LABEL".log
..
SAVE OUTF=bias_$"LABEL"_25.str
```

This will produce files called testcase1.log and bias_testcase1_25.str

SAVE

SAVE — saves all node point information into an output file.

Note: In all cases the region boundaries, electrodes, mesh and doping are saved. If a SOLVE statement has preceded the SAVE statement all electrical data from the last solution is stored.

Syntax

```
SAVE OUTFILE=<filename> [MASTER]
```

Parameter	Type	Default	Units
OUTFILE	Character		
MASTER	Character	True	
PISCES	Logical	False	
PATTERNS	Character		
STRUCTURE	Character	False	

Description

OUTFILE specifies the name of an output file name.

MASTER specifies that the output file will be written in a standard structure format.Files in this format can be plotted in TONYPLOT

STRUCTURE Synonym for save.

PISCES specifies that the output file will be written in the original PISCES-II format.

PATTERNS specifies a character string representing the root of the file names where near and far field patterns are written for LASER. The near field pattern file is appended with the string ".nfp" and the far field pattern file is appended with the string ".ffp".

Basic Save Example

```
SOLVE V1=5
SAVE OUTF=data1.str
```

is equivalent to

```
SOLVE V1=5 OUTF=data1.str MASTER
```

Example with User-defined Output

In the second example, the SAVE and OUTPUT commands are used to produce two output files for the same bias. The OUTPUT statement selects which data will be stored in each file. The first file (data1.str) contains the default contents, total electric field, and components of electron velocity. The second file (data2.str) contains components of hole velocity and band edge potentials. Note that the ^EX.VELO and ^EY.VELO parameters are used to prevent electron velocity components from being stored in file data2.

```
OUTPUT E.FIELD EX.VELO EY.VELO
```

```
SAVE OUTF=data1.str
OUTPUT HX.VELO HY.VELO CON.BAND VAL.BAND ^EX.VELO ^EY.VELO
SAVE OUTF=data2.str
```

Note: Contents of the saved file can be customised using the OUTPUT statement.

SINGLEEVENTUPSET

SINGLEEVENTUPSET — specifies the values of parameters used in Single Event Upset modeling in DEVICE3D or BLAZE3D.

Syntax

```
SINGLEEVENTUPSET <parameters>
```

Parameter	Type	Default	Units
A1	Real		
A2	Real		
A3	Real		
A4	Real		
B1	Real		
B2	Real		
B3	Real		
B4	Real		
B.DENSITY	Real	0	cm^{-3}
BEAM.RADIUS	Real	0	μm
DENSITY	Real		cm^{-3}
ENTRYPOINT	Real	vector	μm
EXITPOINT	Real	vector	μm
PCUNITS	Logical	False	
RADIALGAUSS	Logical	False	
RADIUS	Real		μm
RESCALE	Logical	False	
TFINAL.SEU	Real		
T0	Real		c
TC	Real		c
TF	Real		sec
UNIFORM	Logical	False	

Description

A1, **A2**, **A3** and **A4** are the first set of parameters for the length dependence of the charge generation pulse.

B1, **B2**, **B3** and **B4** are the second set of parameters for the length dependence of the charge generation pulse.

B.DENSITY specifies the number of electron-hole pairs per unit volume or generated charge in pico Coulumbs per micron if the PCUNITS parameter is specified.

BEAM.RADIUS is the radius of the beam where the generation rate is maintained constant. Beyond this point the generation will decay by either an exponential or by a Gaussian function.

DENSITY specifies the number of electron-hole pairs per unit volume generated along the alpha particle track.

ENTRYPOINT specifies the x, y, and z coordinates of the beginning of the alpha particle track. The specified point should belong to the semiconductor region.

EXITPOINT specifies the x, y, and z coordinates of the end of the alpha particle track. The specified point should belong to the semiconductor region.

PCUNITS sets the units of B.DENSITY to be pC per micron.

RADIALGAUSS specifies the Gaussian radial dependence of the charge generation pulse. By default the exponential dependence is used.

RADIUS specifies the radius of the alpha particle track.

RESCALE causes nodal generation rates to be scaled by the ratio of the integral of the analytic generation rate divided by the numerically intergrated value. **NOTE**: this may cause problems when the track radius approaches device dimensions.

TFINAL.SEU specifies the finish time for the track, this defaults to the finish time of the transient simulation if TFINAL.SEU is not specified.

T0 specifies the peak in time of the charge generation pulse.

TC specifies the width of the charge generation pulse.

UNIFORM specifies that a uniform generation rate corresponding to that specified by the DENSITY parameter is applied.

SEU Example

This statement specifies a track path, radius and density:

```
SINGLEEVENTUPSET ENTRYPOINT="1.5,2.0,0.0"\
                 EXITPOINT="1.0,1.0,4.0 RADIUS=0.05" \
                 DENSITY=1E18
```

Note: For user-defined Single Event Generation profiles the C-INTERPRETER function F3.RADIATE on the BEAM statement can be used.

SOLVE

SOLVE — instructs ATLAS to perform a solution for one or more specified bias points.

Syntax

```
SOLVE [<ion>] <dc> [<fp>][<ep>][<tp>][<ac>][<photo>] [<thermal>]
```

Parameter	Type	Default	Units
AC.ANALYSIS	Logical	False	
ANAME	Character		
AR.INDEX	Real	1.0	
AR.THICKNESS	Real	0.0	microns
ASCII	Logical	False	
AUTO	Logical	True	
B<n>	Real	0.0	W/cm^2
CNAME	Character		
COMPLIANCE	Real		
CURVETRACE	Logical	False	
CYCLES	Integer	1	
CYCLIC.BIAS	Logical	False	
CYCLIC.RELAX	Real	0.2	
CYCLIC.TOL	Real	$1.0x10^{-5}$	
DECAY	Real		s
DELTAV	Real	0.1	V
DIRECT	Logical	False	
DT	Real	0	s
E.COMPL	Real		
E.CRIT	Real	$1.0x10^{-8}$	
ELECTRODE	Integer		
ENDRAMP	Real		s
FREQUENCY	Real		Hz
FSTEP	Real	0	Hz
GRAD	Logical	False	
I<n>	Real		A/μm

Parameter	Type	Default	Units
IFINAL	Real		A/μm
IMULT	Logical	False	
INAME	Character		
INDEX.CHECK	Logical	False	
INITIAL	Logical	False	
ION.CRIT	Real	1.0	
ION.ELEC	Integer	see Description	
IONIZINT	Logical	False	
IONLINES	Integer	50	none
IONSTOP	Logical	True	
ISTEP	Real	0.0	A/μm
LAMBDA1	Real		μm
LIT.STEP	Real		W/cm^2
LOCAL	Logical	False	
LRATIO	Real	1.0	
L.WAVE	Real		μm
MASTER	Logical	False	
MAX.INNER	Integer	25	
MLOCAL	Logical	True	
MULT.FREQ	Logical	False	
N.BIAS	Real		V
NAME	Character		
NB1	Real		V
NB2	Real		V
NB3	Real		V
NB4	Real		V
NB5	Real		V
NB6	Real		V
NB7	Real		V<B>
NB8	Real		V
NFSTEPS	Integer	0	

Parameter	Type	Default	Units
NLAYERS	Real	15	
NSTEPS	Integer	0	
ONEFILEONLY	Logical	See Description	
OUTFILE	Character		
P.BIAS	Real		
PB1	Real		V
PB2	Real		V
PB3	Real		V
PB4	Real		V
PB5	Real		V
PB6	Real		V
PB7	Real		V
PB8	Real		V
POWER<n>	Real		W
POWERFINAL	Real		W
PREVIOUS	Logical	False	
PROJECT	Logical	False	
PULSE.WIDTH	Real		s
Q<n>	Real		C/μm
QFACTOR	Real	1.0 or previous value	
QFINAL	Real		C/μm
QSTEP	Real		C/μm
RAMPTIME	Real		s
RAMP.LIT	Logical	False	
SCAN.SPOT	Real		
S.OMEGA	Real	1.0	
SINUAMP.COMP	Real	0.0	
SINUVAR.COMP	Real	0.0	
SOR	Logical	False	
SQPULSE	Logical	False	
SS.LIGHT	Real	0.001	W/cm^2

Parameter	Type	Default	Units
SS.PHOT	Logical	False	
T <n>	Real		K
T.COMP	Real	0	K
T.SAVE	Real	0.0	s
TDELAY	Real	0.0	s
TFALL	Real	0.0	s
TRISE	Real	0.0	s
TSAVE.MULT	Real	1.0	
TERMFINAL	Real		K
TERMINAL	Integer	all contacts	
TOLERANCE	Real	$1.0x10^{-5}$	
TRANS.ANALY	Logical	False	
TSTOP	Real		
TWOFILESONLY	Logical	False	
V<n>	Real		V
VFINAL	Real		V
VSTEP	Real	0.0	V
VSS	Real	0.1	V

Description

Each SOLVE statement must specify an initial bias condition. Once any DC condition has been solved, either a transient or AC analysis may be performed. You may also solve for carrier generation due to incident light under DC, or AC analysis transient conditions.

dc is one or more of the DC bias parameters

fp is one or more of the file parameters

ep is one or more of the initial guess or estimate parameters. Estimate parameters are used to specify how the initial approximation for the solution is to be obtained.

tp is one or more of the transient parameters. These parameters are used to specify data for transient analysis.

ac is one or more of the AC parameters. AC parameters are used to specify data for AC analysis.

ion is a set of the ionization integral parameters.

photo is one or more of the photogeneration parameters. Photogeneration parameters are used to specify illumination data.

therm is one or more of the thermal parameters. Thermal parameters are used for obtaining solutions in THERMAL3D.

DC Parameters

CURVETRACE initiates curve tracing. See also the CURVETRACE statement.

ELECTRODE specifies electrodes that you wish to add voltage and current increments (VSTEP and ISTEP) to. If n electrodes are to be stepped, ELECTRODE should be an n-digit integer, where each of the digits is a separate electrode number. See also NAME.

I<**name**> specifies the applied current for a named electrode. One of several commonly used terminal names should be specified. These names are as follows: **gate, gg, drain, dd, source, bulk, substrate, emitter, ee, collector, cc, base, bb, anode, cathode, fgate, cgate, ngate, pgate, well, nwell, pwell, channel,** and **ground**.No other user defined names are allowed.This parameter is used when current boundary conditions are selected (see the CONTACT statement)

I<**n**> specifies the terminal current for electrode **n**. This parameter is used when current boundary conditions are selected (see the CONTACT statement). Normally, I defaults to the current from the previous bias point.It is more usual to use electrode names rather than numbers. This parameter is superceded by I<name>

IFINAL specifies the final current value for a set of bias increments. If IFINAL is specified, either ISTEP or NSTEPS must be specified.

IMULT specifies that the current (for current boundary conditions) be multiplied by ISTEP rather than incremented.

ISTEP specifies a current increment to be added to one or more electrodes, as specified by the electrode name applied to the NAME parameter. If ISTEP is specified, either IFINAL or NSTEPS must also be specified.

N.**BIAS** specifies fixed electron quasi-Fermi potentials if electron continuity is not being solved. If N.BIAS is not specified, then local quasi-Fermi potentials based on bias and doping are used. However, if FIX.QF is set in the METHOD statement, quasi-Fermi levels will be set to the maximum bias.

NB<**n**> allows region by region specification of N.BIAS . The **n** index corresponds to the region index for which the specified value of electron quasi-fermi level applies.

NAME specifies that the named electrode is to be ramped. Custom electrode names are supported by name. See also the V<name> parameter.

NSTEPS specifies the number of DC bias increments.

P.**BIAS** specifies fixed hole quasi-Fermi potentials if hole continuity is not being solved. If P.BIAS is not specified, then local quasi-Fermi potentials based on bias and doping are used. However, if FIX.QF is set in the METHOD statement, quasi-Fermi levels will be set to the minimum bias.

PB<n> allows region by region specification of P.BIAS. The **n** index corresponds to the region index for which the specified value of hole quasi-fermi level applies.

Q<**name**> specifies the charge on a named electrode. These names are as follows: **gate, gg, fgate, cgate, ngate, pgate**.No other user defined names are allowed.This parameter is used when floating or charge boundary conditions are selected (see the CONTACT statement)

Q<**n**> specifies the charge on electrode number **n**.It is more usual to use electrode names rather than numbers. This parameter is superceded by Q<name>

QFINAL specifies the final charge for a set of bias increments. If QFINAL is specified, either QSTEP or NSTEPS must also be specified.

QSTEP specifies a charge increment to be added to one or more electrodes, as specified by the electrode name applied to the NAME parameter. If QSTEP is specified, either QFINAL or NSTEPS must also be specified.

V<**name**> specifies the bias voltage for a named electrode. One of several commonly used terminal names should be specified. These names are as follows: **gate, gg, drain, dd, source, bulk, substrate, emitter, ee, collector, cc, base, bb, anode, cathode, fgate, cgate, ngate, pgate, well, nwell, pwell, channel,** and **ground**.No other user defined names are allowed.

V<**n**> specifies the bias voltage for electrode **n**. Normally, V**n** defaults to the potential from the previous bias point. It is more usual to use electrode names rather than numbers. This parameter is superceded by V<name>

VFINAL specifies the final voltage for a set of bias increments. If VFINAL is specified, either VSTEP or NSTEPS must also be specified.

VSTEP specifies a voltage increment to be added to one or more electrodes, as specified by the electrode name applied to the NAME parameter. If VSTEP is specified, either VFINAL or NSTEPS must also be specified.

File Output Parameters

MASTER specifies that the output file selected by the OUTFILE parameter will be written in a standard structure file rather than in the older PISCES-II binary format.

ONEFILEONLY specifies that only one filename will be used to save solutions during a bias ramp. If this parameter is specified filename incrementation described under OUTFILE is not applied.This parameter is true by default unless the NAME or ELECTRODE parameters or transient simulationis specified. When CURVETRACE is used it is necessary to explicitly set this parameter to false using ^ONEFILEONLY to save a different filename at each bias point.

OUTFILE specifies the name of the output file in which bias point solution information will be stored. If an electrode is stepped so that more than one solution is generated by the SOLVE statement, the ASCII code of the last non-blank character of the supplied file name will be incremented by one for each bias point in succession, resulting in a unique file per bias point.

Note: The output file specified by OUTFILE has a limit of 132 characters on a UNIX system, and eight characters on a PC system.

Initial Guess Parameters

INITIAL sets all voltages to zero. If this parameter is not specified for the first bias point in a given structure a SOLVE INIT statement is automatically inserted. The SOLVE INIT statement is always solved in zero carrier mode with no external elements attached.It is used to provide a good initial guess to subsequent solutions.

LOCAL specifies that the initial approximation should use local values of quasi-Fermi levels. Refer to the "Initial Guesses" section of the Numerical Techniques Chapter for more detailed information.

MLOCAL specifies the modified local initial guess. This parameter is used when solved for carrier temperatures in the energy balance models. If any energy balance model is specified, MLOCAL defaults to true.

PREVIOUS specifies that the previous solution as the initial approximation.

PROJ specifies that an extrapolation from the last two solutions will be used as an initial approximation. This parameter may be used if there are two or more existing solutions and equivalent bias steps are taken on any electrodes that are changed.

Note: If no initial guess parameters are specified, ATLAS will use PROJ wherever possible.

QFACTOR specifies as scaling factor to improve initial guesses when the QUANTUM model is used. This parameter should be ramped slowly from zero to unity at the start of quantum effect simulations.

Compliance Parameters

Compliance parameters define a current limit for a DC bias ramp or transient simulation. Simulation can be terminated by monitoring the current and checking against a user defined limit.

COMPLIANCE sets a limit on the current from the electrode which has been specified by the CNAME or E.COMPLIANCE parameter. When the COMPLIANCE value is reached, any bias ramp is stopped and the program continues withthe next line of the input file. The COMPLIANCE parameter is normally specified in A. If the GRAD parameter is specified, COMPLIANCE is specified in A/V.

E.COMPLIANCE specifies the electrode number to be used by the COMPLIANCE parameter. See also CNAME.

CNAME specifies the name of the electrode used by the COMPLIANCE.

GRAD specifies that the compliance value is a current/voltage gradient, and not a current value.

Transient Parameters

CYCLES specifies the number of periods to be simulated (both FREQUENCY and TRANS.ANALY must be specified when this parameter is used). (synonym: PERIODS)

CYCLIC.BIAS specifies that a cyclic bias[140] simulation is being performed. Cyclic biasing allows the effects of repeated trapazodial/square transient pulses (see SQPULSE) on the output characteristics to be determined, without having to do the extremely large numbers of cycles. When CYCLIC.BIAS is specified, ATLAS will calculate a series of trapazodial/square pulses using the SQPULSE parameters. At the end of the third cycle (and at all subsequent cycles) the values of the potential (V), electron concentration (n), hole concentration (p) and trap probability of occupation (f_T) will be modified according to the equation: x(k+1) = x(k) - CYCLIC.RELAX*((x(k) - x(k-1)) * dx(k))/(dx(k) - dx(k-1))

where x is a parameter such as V, n, p or f_T at every node point, x(k) is the updated value calculated from the equation, k is the cycle number (with k+1 being the new update for parameter x), CYCLIC.RELAX is the relaxation factor, $\delta x(k)$ is the difference between the simulated values of x at the start and beginning of the current cycle, while $\delta x(k-1)$ is the equivalent for the previous cycle.

Steady state cyclic convergence is determined by comparing the normalizing sum of the updated values of V, n, p and fT with a tolerance value (CYCLIC.TOL).

CYCLIC.RELAX specifies the **CYCLIC.BIAS** relaxation factor (the recommended range for this parameter to ensure stable convergence is between 0.2 and 1).

CYCLIC.TOL specifies the **CYCLIC.BIAS** tolerance factor.

DECAY specifies the time constant used for defining an exponential change in bias for transient simulation.

ENDRAMP applies any bias changes as linear ramps. ENDRAMP specifies the exact end of the ramp in running time; i.e. the ramp will start at $t=t_0$ and end at t=ENDRAMP.

NSTEPS can be used to signal the end of the run; i.e., the final time would be $t = t$ + NSTEPS* DT. This parameter may be used instead of the TSTOP parameter.

PULSE.WIDTH specifies the time constant used for sinusoidal nonlinear time domain simulation. If SQPULSE is specified, PULSE.WIDTH is the width of the trapazodial/square pulse not including the rise and fall times.

RAMPTIME applies any bias changes as linear ramps. RAMPTIME specifies a ramp interval in seconds; i.e., the ramp will begin at $t=t_0$ and ends at $t=t_0$ +RAMPTIME.

SINUAMP.COMP specifies that sinusoidal amplitude compliance will be used. The transient simulation will stop if the amplitude of a sinusoidal waveform is less than the value of SINUAMP.COMP.

SINUVAR.COMP specifies that sinusoidal variance compliance will be used. The transient simulation will stop if the change in amplitude of a sinusoidal waveform is less than SINUVAR.COMP.

SQPULSE specifies that multiple trapazodial/square transient pulses will be applied. The pulse is controlled by the parameters: TDELAY, TRISE, PULSE.WIDTH, TFALL and FREQUENCY.

T.COMP specifies a temperature for temperature compliance. Once the specified temperature is obtained at some location on the mesh the solution process is discontinued.

T.SAVE specifies a time increment at which the device structure files are saved. Note: Actual time steps may not correspond to the user specified increment.

TDELAY specifies the time delay before the first cycle of multiple trapazodial/square transient pulses (SQPULSE) will be applied.

TFALL specifies the fall time for trapazodial/square transient pulses (SQPULSE).

TRISE specifies the rise time for trapazodial/square transient pulses (SQPULSE).

TRANS.ANALY specifies that transient analysis is being performed and that a sinusoid with a frequency, specified by the FREQUENCY parameter, should be applied.

TSAVE.**MULT** specifies a multiplier the save time increment, T.SAVE, is multiplied by after each time a structure is saved.

TSTEP or **DT** specifies the time-step to be used. For automatic time-step runs, DT is used to select only the first time step (see the METHOD statement).

TSTOP specifies the end of the time interval. If simulation begins at $t=t_0$, it will end at $t=TSTOP+t_0$.

AC Parameters

AC.**ANALYSIS** specifies that AC sinusoidal small-signal analysis should be performed after solving for the DC condition. The full Newton method must be used for this analysis. This is typically specified with the statement

```
METHOD NEWTON CARRIERS=2
```

ANAME specifies the name of the electrode to which the AC bias will be applied. See also TERMINAL.If no ANAME is specified, all electrodes have AC bias applied in turn.

AUTO selects an automatic AC analysis method. This method initially uses SOR. The DIRECT method will be used if convergence problems occur. We strongly recommend the use of the AUTO parameter.

DIRECT selects the direct AC solution method. This method is robust at all frequencies, but slow.

FREQUENCY specifies the AC analysis frequency. Analysis may be repeated at a number of different frequencies (without solving for the DC condition again) by specifying FSTEP. FREQUENCY can also be used to specify a sinusoidal or square pulse frequency for transient simulations.

FSTEP specifies a frequency increment which is added to the previous frequency. If MULT.FREQ is specified, the frequency will be multiplied by FSTEP.

MAX.**INNER** specifies the maximum number of SOR iterations.

MULT.**FREQ** specifies that the frequency will be multiplied by FSTEP, instead of added to FSTEP.

NFSTEPS specifies the number of times that the frequency is to be incremented by FSTEP.

S.**OMEGA** specifies the SOR parameter. This parameter is not the AC frequency.

SOR selects the SOR AC solution method. Although SOR is fast, it should only be used when you are performing simulations at low frequencies. Low frequency can be defined here as at least an order of magnitude below the cutoff frequency.

TERMINAL specifies the electrode number to which the AC bias will be applied. Although more than one contact number may be specified (via concatenation), each will be solved separately. Each contact that is specified generates a column of the admittance matrix. If no TERMINAL is specified, all electrodes have AC bias applied in turn. See also ANAME.

TOLERANCE specifies SOR convergence criterion.

VSS specifies the magnitude of the applied small-signal bias. The approach used for small-signal analysis constructs a linear problem based on derivatives calculated during Newton iterations, so adjusing VSS will generally not affect the results.

Ionization Integral Parameters

Ionization parameters are used to calculate ionization integrals. No calculation will take place unless the IONIZINT parameter is specified.

DELTAV since the electric field can be near zero at electrodes, the electric field line calculations begin at a distance from the electrode. A potential contour is drawn around the electrode at a distance where the potential is DELTAV less than the applied bias on the contact. Defaults to 0.1 V, but will typically need to be increased especially for power devices and heavily doped contact regions.

E.CRIT specifies the minimum electric field used to calculate integration integrals. Field lines will terminate when the field drops below E.CRIT.

INAME specifes the electrode name from which the electric field lines are calculated.The default is to use the same as electrode as specified in NAME.

ION.CRIT specifies the critical value of the ionization integral used by IONSTOP to terminate the simulation. When the critical value is reached, any bias ramp will be terminated and the next line of the input file will be executed.

ION.ELEC specifies the electrode from which the electric field lines are calculated. This parameter defaults to the electrode which is being ramped (if any).

IONIZINT enables the calculation of ionization integrals along electric field lines.

IONLINES specifies the number of electric field lines to be calculated. Ionization integrals are calculated along each line.

IONSTOP stops the bias ramp if integral is greater than 1.0

LRATIO specifies the ratio between the starting points of the electric field lines. An LRATIO value of between 0.5 and 1.5 is recommended. LRATIO=1 means that the spacing between the starting points of the electric field lines is equal. LRATIO < 1 means more lines start towards the left hand side of the structure. LRATIO > 1 means more lines start towards the right hand side.

Photogeneration Parameters

All these parameters require LUMINOUS to be licensed for correct operation.

AR.INDEX specifies the index of refraction for an antireflective coating.

AR.THICKNESS specifies the thickness in microns of an antireflective coating.

B<*n*> specifies the optical spot power associated with optical beam number **n**. The beam number must be an integer from 1 to 10. .

BEAM specifies the beam number of the optical beam when AC photogeneration analysis is performed. Unlike the ELECTRODE parameter, this parameter can only be used to specify a single beam.

INDEX.CHECK specifies that the real and imaginary refractive indices used in the ray tracing of each beam will be printed to the run-time output. This parameter can be used as confirmation of the input of user-defined refractive indices or to check the default parameters.

L.WAVE specifies the luminous wavelength to use to calculate the luminous power estimate for radiative recombination. If a positive value of L.WAVE is specified the luminous power will be saved in any log file specified for that solution.

LAMBDA1 specifies the wavelength of the optical source (beam) number 1 for this solution. This parameter can be used to perform analysis of spectral response as a function of wavelength.

LIT.STEP selects the light intensity increment of all optical beams which have been specified. This parameter is used when light intensity varies by stepping (similar to the VSTEP parameter). If LIT.STEP is specified, the NSTEP parameter should be used to select the number of steps.

RAMP.LIT specifies that the light intensity is to be ramped when transient simulations are performed. If RAMP.LIT is specified, transient mode parameters such as RAMPTIME, TSTEP, and TSTOP must also be specified. The RAMP.LIT parameter affects all specified optical beams (i.e., all beams are ramped).

SCAN.SPOT specifies a beam number for spot scanning. Spot scanning requires that the user specify the RAYS parameter of the BEAM statement. with this specification the incident light is split into the user specified number of rays. During the spot scan solutions are obtained with the beam energy applied to each of the rays in sequence.

SS.LIGHT specifies the intensity in the small signal part of the optical beam when small signal AC photogeneration analysis is performed.

SS.PHOT specifies that small signal AC analysis will be performed on the optical beam selected by the BEAM parameter. When SS.PHOT is specified, other AC parameters (e.g., FREQUENCY, FSTEP, MAX.INNER, MULT.FREQ, NFSTEPS, S.OMEGA, and TOLERANCE) should also be specified. It is not necessary to specify the AC.ANALYSIS parameter when performing small signal AC analysis on optical beams.

Thermal3D parameters

See also the section "Obtaining Solutions in THERMAL3D" of this manual.

POWER<**n**> specifies the power in watts on the n-th heat source.

POWERFINAL specifies the final power for a linearly ramped heat source.

T<**n**> specifies the temperature in K on the n-th heat sink.

TEMPFINAL specifies the final temperature for a linearly ramped heat sink.

DC Conditions Example

The following statement solves for a defined bias point and saves the solution to output file OutA. The voltages on electrodes other than the gate will keep the value from the previous SOLVE statement.

```
SOLVE VGATE=0.1 OUTFILE=OutA
```

Bias Stepping Example

In the next example, bias stepping is illustrated. The two SOLVE statements produce the following bias conditions:

Bias Point	Vgate	Vdrain	Vsub
1	0.0	0.5	-0.5
2	1.0	0.5	-0.5
3	2.0	0.5	-0.5
4	3.0	0.5	-0.5
5	4.0	0.5	-0.5
6	5.0	0.5	-0.5

The solutions for these bias points will be saved to the files OUT1, OUTA, OUTB, OUTC, OUTD and OUTE. The initial approximation for each bias point is obtained directly from the preceding solution.

For bias points 4, 5, and 6, the program will use a PROJ to obtain an initial approximation. Since, starting with bias point 4, both of its preceding solutions (bias points 2 and 3) only had the same electrode bias (number 1) altered.

```
SOLVE Vdrain=.5 Vsub=-.5 OUTF=OUT1

SOLVE Vgate=1 VSTEP=1 VFINAL=5  NAME=gate OUTF=OUTA
```

Transient Simulation Example

The following sequence is an example of a time dependent solution. The METHOD statement specifies second-order discretization, automatic time-step selection, and an automated Newton-Richardson procedure.

The first SOLVE statement then computes the solution for a device with 1V on the base electrode and 2V on the collector in steady-state. The second SOLVE statement specifies that the base electrode is to be ramped to 2V over a period of 10 ns and is left on until 25 ns. Each solution is written to a file. The name of the file is incremented in a manner similar to that described for a DC simulation (UP1, UP2, etc.). Note that an initial time step had to be specified in this statement.

The third SOLVE statement ramps ther base down from 2V to - 0.5V in 20 ns (end of ramp is at =45 ns). The device is then solved at this bias for another 55 ns (out to 100 ns). Each solution is again saved in a separate file (DOWN1, DOWN2, etc.).

No initial timestep was required since one had been estimated from the last transient solution from the previous SOLVE statement.

Finally, the fourth SOLVE statement performs the steady-state solution at Vbe = -0.5V and Vce = 2V.

```
METHOD 2ND TAUTO AUTONR
SOLVE  Vbase=1 Vcollector=2
SOLVE  Vbase=2 DT=1E-12 TSTOP=25E-9 RAMPTIME=10E-9 OUTF=UP1
SOLVE  Vbase= –0.5 TSTOP=100E-9 RAMPTIME=20E-9 OUTF=DOWN1
SOLVE  Vbase= –0.5 Vcollector=2
```

AC Analysis Example

The following example illustrates an application of the SOLVE statement for AC analysis. It is assumed that the device has three electrodes. This analysis is being performed after DC conditions are solved at V1 = 0V, 0.5V, 1.0V, 1.5V, and 2.0V. A series of 10mV AC signals with frequencies of 1 MHz, 10 MHz, 100 MHz, 1 GHz, 10 GHz, and 100 GHz are applied to each electrode in the device.

Ninety AC solutions will be performed (5 X 6 X 3 = 90).

```
SOLVE V1=0 V2=0 V3=0 VSTEP=0.5 NSTEPS=4 ELECT=1 \
AC FREQ=1E6 FSTEP=10 MULT.F NFSTEP=5 VSS=0.01
```

Photogeneration Examples

The following statement simulates two DC optical beams incident on the device with optical spot powers of 15 and 25 W/cm^2.

```
SOLVE B1=15 B3=25
```

The next example shows how DC spot power of the two optical beams can be stepped simultaneously. Beam #1 will increase to 35 W/cm^2 and beam #3 will increase to 45 W/cm^2.

```
SOLVE LIT.STEP=5.0 NSTEP=4
```

In the next example, the beams are ramped in the time domain. Beam #1 is ramped down to 0 W/cm^2 and beam #3 is ramped up to 100 W/cm^2. The duration of the ramp is 1 ns. After the ramp, simulation will continue for 4 ns.

```
SOLVE B1=0 B3=100 RAMP.LIT TSTEP=2E-11 RAMPTIME=1E-9 TSTOP=5E-9
```

Next, the small signal response of a single beam is analyzed. First, **ATLAS** will solve the DC characteristics at the specified optical spot powers. Then, the AC response of beam #1 will be calculated at a frequency of 10 MHz.

```
SOLVE B1=10 B3=20 BEAM=1 FREQUENCY=1e7 SS.PHOT SS.LIGHT=0.01
```

Finally, frequency stepping is used to look at the small signal AC frequency response of one of the beams. AC response is calculated at frequencies from 1kHz to 100MHz at each decade. The MULT.F parameter is used to geometrically increase the frequency.

```
SOLVE B1=10 B3=5 BEAM=1 SS.PHOT SS.LIGHT=0.01 \
      MULT.F FREQUENCY=1E3 FSTEP=10 NFSTEP=6
```

Ionization Integral Example

Ionization integrals are used to estimate the breakdown voltage from analysis of the electric field. They can be used in zero carrier mode providing much faster simulation that conventional breakdown analysis. The ionization syntax uses the parameter IONIZ to enable the ionization integral calculation. An equipotential contour is calculated at a potential DELTAV from the contact NAME (or INAME). Electric field lines are started at distances along this potential contour. IONLINES sets the number of lines and LRATIO is the ratio of the distance between the starting points of the lines. The following syntax is for a PMOS transistor. See the on-line examples for other cases using ionization integrals.

```
IMPACT SELB

METHOD CARR=0

SOLVE VDRAIN=-1 VSTEP=-1 VFINAL=-20.0 NAME=DRAIN \
       IONIZ IONLINES=50 LRATIO=0.9 DELTAV=1.2
```

Note: There are over 300 on-line examples supplied with ATLAS to provide examples of sequences of SOLVE statements applied to practical problems for a variety of device technologies.

SPREAD

SPREAD — distorts rectangular grids defined by ATLAS in the vertical direction to follow surface and junction contours.

Note: The use of this parameter is **not recommended** by SILVACO.

Syntax

```
SPREAD LEFT|RIGHT WIDTH=<r> UPPER=<i> LOWER=<i>
Y.LOWER|THICKNESS [<options>]
```

Parameter	Type	Default	Units
ENCROACH	Real	1.0	
GRADING	Real	1.0	
GR1	Real	1.0	
GR2	Real	1.0	
LEFT	Logical	False	
LOWER	Integer		
MIDDLE	Integer	Halfway between UPPER and LOWER	
RIGHT	Logical	False	
THICKNESS	Real		μm
UPPER	Integer		
VOL.RATIO	Real	0.44	
WIDTH	Real		μm
Y.LOWER	Real		μm
Y.MIDDLE	Real	0.50	μm

Description

SPREAD can reduce the grid complexity of specific simulations. Since the SPREAD statement is somewhat complicated, we suggest that you follow the supplied examples very carefully until you are confident that you understand the workings of this statement.

Mandatory Parameters

LEFT distorts the left side of the grid. If LEFT is specified, RIGHT must not be specified.

LOWER specifies the lower y-grid line above which distortion will take place.

RIGHT distorts the right side of the grid. If RIGHT is specified, LEFT must not be specified.

THICKNESS specifies the thickness of the distorted region. Unless VOL.RATIO is set to 0 or 1, THICKNESS will usually move the positions of both the UPPER and LOWER grid lines. The Y.LOWER and THICKNESS parameters define the distorted grid region. Only one of these parameters should be specified.

UPPER specifies the upper y-grid line under which distortion will take place.

WIDTH specifies the width from the left or right edge (depending on whether LEFT or RIGHT is selected) of the distorted area. The actual **x** coordinate specified by WIDTH (min[*x*]+ WIDTH for LEFT, max[*x*] + WIDTH for RIGHT) will lie in the middle of the transition region between distorted and undistorted grid regions.

Y.LOWER specifies the physical location in the distorted region to which the line specified by LOWER will be moved. The line specified by UPPER is not moved. The Y.LOWER and THICKNESS parameters define the distorted grid region. Only one of these paramaters should be specified.

Optional Parameters

ENCROACH defines the abruptness of the transition between a distorted and nondistorted grid. The transition region becomes more abrupt with smaller ENCROACH factors (the minimum is 0.1).

Note: Depending on the characteristics of the undistorted grid, long. thin, or obtuse triangles may result if too low an ENCROACH value is used.

GRADING specifies a grid ratio which produces a non-uniform grid in the distorted region. This parameter is identical to the RATIO parameter in the X.MESH and Y.MESH statements.

GR1 and **GR2** may be used instead of the GRADING parameter. GR1 and GR2 may be specified in conjunction with MIDDLE (y grid line) and Y.MIDDLE (location) so that GR1 specifies grading in the spread region from UPPER to MIDDLE, and GR2 specifies grading from MIDDLE to LOWER.

MIDDLE specifies the y-grid line that serves as a boundary between the grading specified by GR1 and the grading specified by GR2.

VOL.RATIO specifies the ratio of the downward displacement of the lower grid line to the net increase in thickness. The default (0.44) should be used for oxide-silicon interfaces. VOL.RATIO is ignored if Y.LOWER is specified.

Y.MIDDLE specifies the physical location in the distorted grid to which the line specified by MIDDLE will be moved.

Examples

This example spreads a uniform 400 Å of oxide to 1000 Å on the left side of the device. This increases oxide thickness by 600 Å. Because VOL.RATIO is not specified, the default (0.44) is used. Therefore, 0.44 X 600 = 264 Å of the net increase will lie below the original 400 Å and 0.56 X 600 = 336 Å of the net increase will lie above the original 400 Å. The width of the spread region is 0.5 μm, and the oxide taper is quite gradual because of the high encroachment factor. The grid is left uniform in the spread region.

```
# *** Mesh definition ***
 MESH NX=30 NY=20 RECT
 X.M N=1  L=0
 X.M N=30 L=5
 Y.M N=1  L=-.04
 Y.M N=5  L=0
 Y.M N=20 L=1 R=1.4
 # *** Thin oxide ***
 REGION IY.H=5 OXIDE NUM=1
```

```
# *** Silicon substrate ***
REGION IY.L=5 SILICON NUM=2
# *** Spread ***
SPREAD LEFT W=0.7 UP=1 LO=4 THICK=0.1 MID=2 Y.MID=0.05
```

In the second example, the right side of the grid is distorted in order to follow a junction contour. The initial grid is assumed to be above. Y.LOWER is used so that there is no increase in the size of the device, just grid redistribution. When Y.LOWER is set to the junction, the ENCROACH parameter should be chosen such that the lower grid line (LOWER=10) follows the junction as closely as possible. The grid is graded so that grid lines are spaced closer together as they approach the junction. Because the point specified by WIDTH lies in the middle of the transition region, it should be chosen to be slightly larger than the width of the doping “box.”

```
# *** Doping ***
  DOPING UNIFORM N.TYPE CONC=1E15
  DOPING GAUSS P.TYPE X.LEFT=1.5 X.RIGHT=2 \
  PEAK= CONC=1.e19  RATIO=.75 JUNC=0.3
  # *** Spread ***
  SPREAD RIGHT W=0.7 UP=1 LO=4 THICK=0.3 MID=2 Y.MID=0.10
```

SYMBOLIC

Note: In versions 3.0 and greater, the SYMBOLIC statement is no longer needed. All functions are moved to the METHOD statement.

SYSTEM

SYSTEM — allows execution of any UNIX command within an input file

---

Note: The SYSTEM statement is executed by DECKBUILD and is fully documented in the VWF INTERACTIVE TOOLS MANUAL VOLUME ONE

Note: The SYSTEM command must be enabled using an option on the DECKBUILD Main_Control menu under **Category: Options.**

Examples

The following command will remove all files `test*.str` before a `SOLVE` statement where the `OUTF` parameter is used

```
system \rm -rf test*.str
SOLVE  .... OUTF=test0
```

The `system` command and the UNIX commands are case sensitive.

UNIX commands may be concatentated on a single line using the semicolon (;) operator. For example to run a third party program that reads and writes Silvaco format files with fixed names `input.str` and `output.str`.

```
SAVE OUTF=mysave.str
system mv mysave.str input.str; source myprog.exe; mv output.str myrestart.str
EXTRACT INIT INF=myrestart.str
```

The UNIX re-direct symbol > is not supported by the `system` command. The UNIX `echo` and `sed` syntax can be used instead to output values or variables to a given filename. For example to save the extracted value of variable `$myvariable` to the file `myfile`.

```
system echo $"myvariable" | sed -n " w myfile"
```

THERMCONTACT

THERMCONTACT — specfies the position and properties of thermal contacts. This statement must be used when lattice heating solutions are specified using GIGA or GIGA3D.

Syntax

```
THERMCONTACT NUMBER=<n> <position> [EXT.TEMPER=<n>] [ALPHA=<n>]
```

Parameter	Type	Default	Units
ALPHA	Real	∞	W/(cm2·K)
DEVICE	Character		
ELEC.NUMBER	Integer		
EXT.TEMPER	Real	300	K
NAME	Character		
NUMBER	Integer	1	
X.MAX	Real	Right side of structure	μm
X.MIN	Real	Left side of structure	μm
Y.MAX	Real	Bottom of structure	mμ
Y.MIN	Real	Top of structure	μm
Z.MIN	Real	Front	microns
Z.MAX	Real	Back	microns

Description

At least one thermal contact statement must be specified when simulating lattice heating effects (MODELS LAT.TEMP). The THERMCONTACT statement must appear in the input deck before any METHOD statement.

position is a set of the position parameters described below. Either X.MIN, X.MAX, Y.MIN, and Y.MAX parameters are used to specify the exact postion of the contact or the ELEC.NUMBER parameter to specify an electrode number that the thermal contact coincides with..

NUMBER specifies a thermal contact number from 1 to 20. Contact numbers should be specified in increasing order. This parameters **must** be specified on all THERMCONTACT statements.

ALPHA specifies the reverse value of thermal resistance ($\alpha=1/R_{TH}$). This parameter may not be used if ELEC.NUMBER parameter has been specified.

EXT.**TEMPER** specifies the external temperature.

Synonym: TEMPERATURE

Position Parameters

NAME specifies which region the THERMCONTACT statement applies to. Note that the name must match the name specified in the NAME parameter of the REGION statement.

DEVICE specifies which device in MIXEDMODE simulation the THERMCONTACT statement applies to.

ELEC.**NUMBER** specifies an electrode number that the thermal contact is coincident with.

X.MIN specifies the left edge of the contact.

X.MAX specifies the right edge of the contact.

Y.MIN specifies the top edge of the contact.

Y.MAX specifies the bottom edge of the contact.

Z.MIN specifies the location of the front edge of the thermal contact.

Z.MAX specifies the location of the rear edge of the thermal contact.

Coordinate Definition Example

A thermal contact is located where y-coordinate values range from 10 μm to the bottom side of the structure and x-ccordinate values range from the left edge of the structure to the right edge of the structure (be default). The external temperature is set to 300K and a thermal resistance of 1 is added. Thus, the temperature at y = 10 μm will be greater than 300K once lattice heating effects occur.

```
THERMCONTACT NUM=1 Y.MIN=10 EXT.TEMP=300 ALPHA=1
```

Example Setting Thermal and Electrical Contacts Coincident

The next statement line creates a thermal contact at the location of electrode #4. An external temperature of 400K is specified.

```
THERMCONTACT NUM=2 ELEC.NUM=4 EXT.TEMP=400
```

Note: Location and Parameters of thermal contacts are not stored in the ATLAS solution files. Therefore THERMCONTACT statements must be defined in each ATLAS run involving lattice heating.

TONYPLOT

TONYPLOT — starts the graphical post-processor TONYPLOT

Note: The TONYPLOT statement is executed by DECKBUILD and is fully documented in the VWF INTERACTIVE TOOLS MANUAL VOLUME ONE

Examples

All graphics in ATLAS is performed by saving a file and loading the file into TONYPLOT. The command:

```
TONYPLOT
```

causes ATLAS to automatically save a structure file and plot it in TONYPLOT. The tonyplot window will appear displaying the material boundaries. Use the `Plot:Display` menu to see more graphics options.

This command will display the file `myfile.str`

```
tonyplot -st myfile.str
```

This command will overlay the results of `myfile1.str` and `myfile2.str`

```
tonyplot -overlay myfile1.str myfile2.str
```

Note: For documentation of the extensive features of TONYPLOT for graphical display and analysis consult the TONYPLOT chapter of the VWF INTERACTIVE TOOLS MANUAL VOLUME ONE

TITLE

TITLE — specifies the title (up to 29 characters) that will appear in the standard output. If used the TITLE command should be the first statement inthe ATLAS input file.

Syntax

```
TITLE <string>
```

Example

This example causes the text " *** CMOS p-channel device *** " to be printed at the top of all ATLAS printouts, and screen displays.

```
TITLE *** CMOS p-channel device ***
```

Note: TITLE cannot be used with the automatic ATHENA to ATLAS interface feature of DECKBUILD

TRAP

TRAP — activates bulk traps at discrete energy levels within the bandgap of the semiconductor and sets their parameter values.

Syntax

```
TRAP DONOR|ACCEPTOR E.LEVEL=<r> DENSITY=<r> DEGEN=<v> <capture parameters>
```

Parameter	Type	Default	Units
ACCEPTOR	Logical	False	
DEVICE	Character		
DEGEN.FAC	Real	undefined	
DENSITY	Real		cm^{-3}
DONOR	Logical	False	
E.LEVEL	Real		eV
FAST	Logical	False	
REGION	Integer	0	
SIGN	Real		cm^2
SIGP	Real		cm^2
TAUN	Real		s
TAUP	Real		s

Description

DEVICE specifies which device the statement applies to in MIXEDMODE simulation.

DONOR specifies a donor-type trap level.

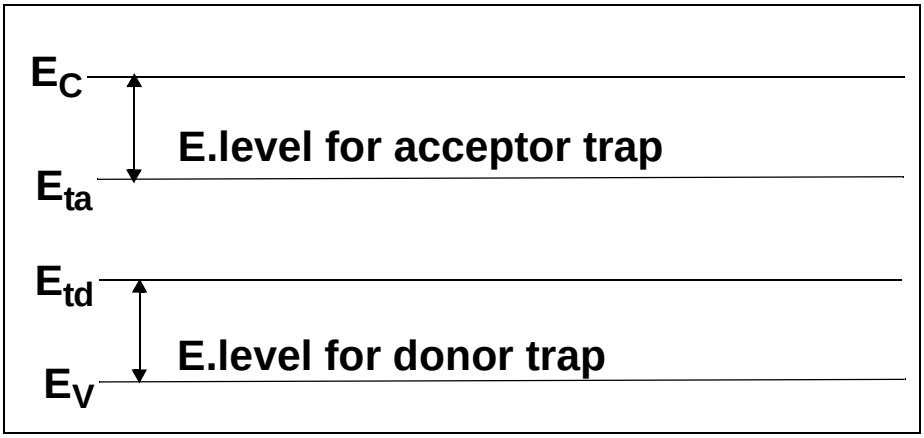
ACCEPTOR specifies an acceptor-type trap level.

REGION specifies which region the traps apply to. If unspecified, the traps apply to all regions.

DENSITY sets the maximum density of states of the trap level.

DEGEN.**FAC** specifies the degeneracy factor of the trap level used to calculate the density.

E.**LEVEL** sets the energy of the discrete trap level. For acceptors E.LEVEL is relative to the conduction band edge, for donors it is relative to the valence band edge.



Capture Parameters

Either the cross section or lifetime parameters should be used to define the capture parameters.

SIGN speccifies the capture cross section of the trap for electrons.

SIGP specifies the capture cross section of the trap for holes.

TAUN specifies the lifetime of electrons in the trap level.

TAUP specifies the lifetime of holes in the trap level.

Multiple Trap Level Definition Example

The following example sets three discrete trap levels within the silicon bandgap. These trap levels will capture carriers, slowing the switching speed of any device. In this example the capture cross sections are used to define the properties of each trap.

```
trap e.level=0.49 acceptor density=2.e15 degen=12 \
     sign=2.84e-15 sigp=2.84e-14
trap e.level=0.41 acceptor density=1.e15 degen=12 \
     sign=7.24e-16 sigp=7.24e-15
trap e.level=0.32 donor density=1.e15 degen=1 \
     sign=1.00e-16 sigp=1.00e-17
```

Note: For distributed trap levels, see the DEFECT statement.

Note: For interface traps see the INTTRAP Statement .

Note: Spatial TRAP distributions in x, y or z directions must be defined on the DOPING statement

UTMOST

UTMOST — converts data from a ATLAS logfile format into an UTMOST logfile format.

Note: UTMOST *12.3.4 or later can read* **ATLAS** *format logfiles directly in batch mode. This statement is obsolete and its use is not recommended.*

Syntax

```
UTMOST BIP|DIODE|MOS <input> OUTFILE=<fn> WIDTH=<n> [<elec>] <cntrl>
 [TRANSLATE INFILE1=<filename>][<analysis>] [<parasites>]
```

Parameter	Type	Default	Units
AC	Logical	False	
ANODE	Integer	1	
APPEND	Logical	False	
BASE	Integer		
BIP	Logical	False	
BULK	Integer		
CATHODE	Integer		
COLLECTOR	Integer		
DEVICE	Integer	1	
DIODE	Logical	False	
DRAIN	Integer		
EMITTER	Integer		
GATE	Integer		
INFILE	Character		
INTERNAL	Logical	False	
LENGTH	Real	1.0	μm
MESFET	Logical	False	
MINSTEP	Real	0.01	V
MOS	Logical	False	
OUTFILE	Character		
POLARITY	Integer	1 (n-type)	
ROUTINE	Integer	1	
SOURCE	Integer		

Parameter	Type	Default	Units
TEMPERATURE	Real	300	K
WELL	Integer		
WIDTH	Real	1.0	µm

Description

You must specify a technology parameter (BIP, DIODE, or MOS), an input file (INFILE1=), and an output file (OUTFILE=). You may specify one or more of the optional **electrode parameters**

The **INFILE** parameter is used to convert up to nine ATLAS logfiles into a single UTMOST logfile. The ATLAS logfiles must be specified in the form:

```
INFILE1=<fn> INFILE2=<fn> INFILE3=<fn> . . .
```

where **fn** is the name of the logfile that you wish to convert.

OUTFILE specifies the name of a file in which I-V data will be stored in an UTMOST file format.

APPEND specifies that I-V data should be appended to the output file specified by OUTFILE2.

WIDTH is used to specify the width of the device. Electrode current is multiplied by the value of WIDTH before being saved in the logfile.

Technology Parameters

BIP creates an UTMOST bipolar transistor log file.

DIODE creates an UTMOST diode log file.

MOS creates an UTMOST MOSFET log file.

MESFET creates an UTMOST MESFET log file.

Electrode Parameters

Different electrode parameters may be specified with different technologies.

BIP Technology

BASE specifies the base electrode number.

COLLECTOR specifies the collector electrode numb.

EMITTER specifies the emitter electrode numb.

POLARITY indicates the device type. Polarity=1 specifies npn. Polarity=-1 specifies pnp.

MOS And **MESFET** Technologies

Note: If you have used the NAME parameter of the ELECTRODE statement to assign standard electrode names (bulk, drain, gate, and source), it is not necessary to re-specify these names in the UTMOST statement.

BULK or **SUBSTRATE** specifies the bulk electrode number.

DRAIN specifies the drain electrode number.

GATE specifies the gate electrode number.

POLARITY indicates the device type. POLARITY=1 specifies nmos. POLARITY=-1 specifies pmos.

SOURCE specifies the source electrode number.

Control Parameters

The optional control parameters are used to specify what type of information will be converted to an UTMOST logfile.

AC specifies that input logfiles contain AC parameters and that the UTMOST routine numbers refer to the UTMOST capacitance routines.

DEVICE specifies the device (or row) number used to identify different devices in UTMOST.

Synonym: ROW

INTERNAL specifies that internal contact voltage will be used instead of applied bias.

LENGTH specifies the length of the device.

MINSTEP specifies the minimum voltage step between data points.

TEMPERATURE specifies the simulation temperature.

ROUTINE specifies which UTMOST routine number data will be saved for. The following routines are supported.

BIP Technology

ROUTINE=1 specifies I_C vs V_{CE} curves. Each input file holds I-V data for a solution with a fixed base current and V_{CE} stepped. (UTMOST IC/VCE routine).

ROUTINE=9 specifies a B_F vs V_{BE} plot. Each input file holds I-V data for a solution with constants V_{CE} and V_{BE} stepped. (UTMOST BFvsIC routine)

ROUTINE=10 specifies a B_F vs V_{BC} plot. Each input file holds I-V data for a solution with constants V_{CE} and V_{BC} stepped. (UTMOST BR routine)

ROUTINE=14 specifies a I_C,I_B vs V_{BE} (Gummel) plot. Each input file holds I-V data for a solution with constant V_{CE} and V_{BE} stepped. (UTMOST gummel routine)

ROUTINE=15 specifies a I_E,I_B vs V_{BC} (Reverse Gummel) plot. Each input file holds I-V data for a solution with constants V_{EC} and V_{BC} stepped. (UTMOST rgummel routine)

ROUTINE=29 specifies I_E vs V_{EC} plot. Each input file holds I-V data for a solution with a fixed base current and V_{EC} stepped. (UTMOST IE/VEC routine)

MOS Technology

ROUTINE=1 specifies a I_D vs V_{DS} plot. V_B is constant and V_G is stepped. (UTMOST ID/VD-VG routine)

ROUTINE=2 specifies a I_D vs V_{GS} plot. V_D is constant and V_B is stepped. (UTMOST ID/VG-VB routine)

ROUTINE=3 specifies the Leff,Rsd routine. Two devices of different lengths are needed. The I_D/V_{GS} data should be used.

ROUTINE=5 specifies a I_D vs V_{GS} plot. V_D is constant and V_B=0. (UTMOST VTH routine)

ROUTINE=9 specifies a I_D vs V_{GS} plot. V_D is constant and V_B is stepped. (UTMOST NSUB routine).

ROUTINE=10 specifies the IS routine. The forward diode characteristics of the drain to bulk junction are required.

ROUTINE=11 specifies the LAMBDA routine. An I_D/V_D curve is required.

ROUTINE=**12** specifies the ETA routine. A set of I_D/V_{GS} data for different VDS is required.

ROUTINE=**13** specifies the VMAX routine. A set of I_D/V_{GS} curves for small changes in V_{DS} are required.

ROUTINE=**14** specifies the U0 routine. A set of I_D/V_{GS} data is required.

ROUTINE=**26** specifies a I_D vs V_{GS} plot. V_B is constant and V_D is stepped. (UTMOST ID/VG-VD routine).

MOS Capacitances

The following routines may be specified if both MOS and AC parameters are selected.

ROUTINE=**1** specifies the CGSO routine. C_{GS} vs. V_{GS} data is required.

ROUTINE=**2** specifies the CGDO routine. C_{GD} vs. V_{GS} data is required.

ROUTINE=**5** specifies the CJ routine. C_{BD} vs. V_{DS} data is required.

ROUTINE=**6** specifies the CJSW routine. C_{BD} vs. V_{DS} data is required.

ROUTINE=**12** specifies the CJ/CJSW routine. C_{GD} vs. V_{GS} data is required for two different size drain areas.

Diode Technology

ROUTINE=**1** specifies an I vs V plot. V is stepped. (UTMOST ID vs. VD routine)

Note: UTMOST 12.3.4 or later can read ATLAS format logfiles directly in batch mode. This statement is obsolete and its use is not recommended.

X.MESH, Y.MESH, Z.MESH

<**n**>.MESH — specifies the location of grid lines along the <**n**>-axis in a rectangular mesh for 2-D or 3-D simulation.

Note: The commands are equivalent in the x, y or z directions.

Syntax

```
X.MESH LOCATION=<l> (NODE=<n> [RATIO=<r>])| SPACING=<v>
```

Parameter	Type	Default	Units
LOCATION	Real		μm
NODE	Integer		
RATIO	Real	1	
SPACING	Real		μm

Description

LOCATION specifies the location of the grid line.

NODE specifies mesh line index. There is a limit of 120 mesh lines. These mesh lines must be assigned in increasing order.

RATIO specifies the ratio to use when interpolating grid lines between given locations. Spacing between adjacent gridlines will increase or decrease by the factor assigned to the RATIO parameter. A RATIO value of between 0.667 and 1.5 is recommended. RATIO should not be used if SPACING is specified.

SPACING specifies the mesh spacing at the mesh locations specified by the LOCATION parameter. If the SPACING parameter is specified, the NODE and RATIO parameters should not be specified. If the SPACING parameter is used to specify mesh spacings, the NX, NY and NZ parameters of the MESH statement should not be specified. When the mesh spacings are specified using the SPACING parameter, the mesh size will be calculated.

Example Setting Fine Grid at A Junction

This example shows how to space grid lines closely around a junction at y=0.85 microns.

```
Y.MESH  LOC=0.0  SPAC=0.2
Y.MESH  LOC=0.85 SPAC=0.01
Y.MESH  LOC=2    SPAC=0.35
```

Appendix A: C-Interpreter Functions

Introduction

ATLAS has a built-in C language interpreter that allows many of the models contained in ATLAS to be modified. In order to use the SILVACO C-INTERPRETER, C language functions containing analytic descriptions of the model must be written. The C-INTERPRETER uses the ANSI standard definition of C. If you are not familiar with the C language, we suggest you refer to any of the popular C language references such as the one written by Kernighan and Ritchie [74]. Additional information about the C-INTERPRETER can be found in the SILVACO C-INTERPRETER MANUAL.

The function arguments of the C-INTERPRETER functions and return values are fixed in ATLAS. Thus care should be taken to ensure that the arguments and the return values for the functions match those expected by ATLAS. To help the user, this release of ATLAS includes a set of templates for the available functions. The C-INTERPRETER function template can be obtained by typing:

```
atlas -T filename
```

where *filename* is the name of a file where you want the template to be copied. The C-INTERPRETER function template can also be obtained by accessing the template file through DECKBUILD.

This template file should be copied and edited when implementing user defined C-INTERPRETER functions. In order to use the C-INTERPRETER function, the file name containing this function must be specified in the appropriate ATLAS statement. The relevant statement names and parameters are listed in the template file.

The following example shows how the C-INTERPRETER function "*munsat*" can be used to describe velocity saturation for electrons.

First, examine the template file and find the template for "*munsat*". The template should look something like this:

```
/*
* Electron velocity saturation model.
* Statement: MATERIAL
* Parameter: F.MUNSAT
*/
/* e         electric field (V/cm) */
/* v         saturation velocity (cm/s) */
/* mu0       low field mobility (cm^2/Vs) */
/* *mu       return: field dependent mobility (cm^2/Vs) */
/* *dmde     return: derivative of mu with e */

int munsat(double e, double v, double mu0, double *mu, double *dmde)
{
      return(0);               /* 0 - ok */
}
```

Here we see that the function is passed the electric field, the saturation velocity and the low field mobility. The function returns the field dependent mobility and the derivative of mobility with respect to electric field.

Note: It is important to properly calculate and return derivative values when specified, since the convergence algorithms in ATLAS require these derivatives.

The return value is an error flag. In this case it is set to zero, indictating that the function was succesful (0=OK, 1=fail).

In this example the electron saturation velocity characteristic is changed to one which is akin to that used for holes in silicon. This model could be specified by setting the B.ELE parameter to 1 but for example purposes, the C-INTERPRETER function "*munsat*" is used instead. This will enable comparison between the built-in model and the C-INTERPRETER model.

To implement the model, two lines additional lines of C code in the body of the function are specified as follows:

```
/*
* Electron velocity saturation model.
* Statement: MATERIAL
* Parameter: F.MUNSAT
*/
/* e         electric field (V/cm) */
/* v         saturation velocity (cm/s) */
/* mu0       low field mobility (cm^2/Vs) */
/* *mu       return: field dependent mobility (cm^2/Vs) */
/* *dmde     return: derivative of mu with e */

int munsat(double e, double v, double mu0, double *mu, double *dmde)
{
      *mu  = mu0/(1.0+mu0*e/v);
      *dmde = (*mu)*(*mu)/v;
      return(0);               /* 0 - ok */
}
```

The function then needs to be stored in a file. For example, the function may be stored in the file "*test.lib*". The function is then introduced into a specific example by specifying the file name on the F.MUNSAT parameter in the MATERIAL statement as follows:

```
MATERIAL F.MUNSAT="test.lib"
```

When the input deck is executed the user's C-INTERPRETER functions will be used in place of the built in function. When trying this example, it is useful to place print statements (using the 'printf' C command) in the function to check that the function is working correctly.

A complete list of all the interpreter functions avialable in ATLAS is given in table 15-1 below.

Table 15-1. Complete list of available C-Interpreter functions in ATLAS

Statement	Parameter	Template	Description
MATERIAL	F.MUNSAT	munsat ()	Electron velocity saturation model.

Table 15-1. Complete list of available C-Interpreter functions in ATLAS			
Statement	**Parameter**	**Template**	**Description**
MATERIAL	F.MUNSAT	mupsat ()	Hole velocity saturation model.
MATERIAL	F.BANDCOMP	bandcomp ()	Temperature and composition dependent band parameters.
DOPING	F.DOPING	doping ()	Position dependent net doping.
DOPING	F3.DOPING	doping3()	Position dependent net doping.
DOPING	F.COMPOSIT	composition ()	Position dependent composition fractions.
MATERIAL	F.EPSILON	epsilon ()	Composition and temperature dependent permittivity.
MATERIAL	F.FERRO	ferro ()	Position and field dependent permittivity.
MATERIAL	F.VSATN	vsatn ()	Composition and temperature dependent electron saturation velocity.
MATERIAL	F.VSATP	vsatp ()	Composition and temperature dependent hole saturation velocity.
MATERIAL	F.RECOMB	recomb ()	Position, temperature, and concentration dependent recombination.
MATERIAL	F.INDEX	index ()	Wavelength dependent complex index of refraction.
BEAM	F.REFLECT	reflect ()	Reflection coefficient.
MATERIAL	F.BGN	bgn ()	Composition, temperature, and doping dependent band gap narrowing.
MATERIAL	F.CONMUN	conmun ()	Composition, temperature, and doping dependent electron mobility.
MATERIAL	F.CONMUP	conmup ()	Composition, temperature, and doping dependent hole mobility.
BEAM	F.RADIATE	radiate ()	Generation rate as a function of position.
BEAM	F3.RADIATE	radiate3 ()	Generation rate as a function of position (3D).
MATERIAL	F.TAUN	taun ()	Electron SRH lifetime as a function of position.

Table 15-1. Complete list of available C-Interpreter functions in ATLAS

Statement	Parameter	Template	Description
MATERIAL	F.TAUP	taup ()	Hole SRH lifetime as a function of position.
MATERIAL	F.COPT	copt ()	Radiative recombination rate as a function of composition and temperature.
MATERIAL	F.GAUN	gaun ()	Electron Auguer rate as a function of composition and temperature.
MATERIAL	F.GAUP	gaup ()	Hole Auger rate as a function of composition and temperature.
INTERFACE	F.QF	int_fixed_charge ()	Interface fixed charge as a function of position.
DEGREDAT	F.NTA	devdeg_nta ()	Interface acceptor trap density as a function of location.
DEGREDAT	F.NTD	devdeg_ntd ()	Interface donor trap density as a function of location.
DEGREDAT	F.SIGMAE	devdeg_sigmae ()	Interface electron c apture cross-section.
DEGREDAT	F.SIGMAH	devdeg_sigmah ()	Interface hole capture cross-section.
INTTRAP	F.DENSITY	inte_acc_trap ()	Interface (acceptor/donor) trap density as a function of location.
DEFECT	F.TFTDON	tftdon ()	TFT donor trap densit yas a function of energy.
DEFECT	F.TFTACC	tftacc ()	TFT acceptor trap density as a function of energy.
IMPACT	F.EDIIN	ediin ()	Electron temperature coefficients for impact ionization.
IMPACT	F.EDIIP	ediip ()	Hole temperature coefficients for impact ionization.
MATERIAL	F.TAURN	taurn ()	Energy dependent electron relaxation time.
MATERIAL	F.TARUP	taurp ()	Energy dependent hole relaxation time.
MODELS	F.KSN	fksn ()	Energy dependent electron Peltier coefficient.

Table 15-1. Complete list of available C-Interpreter functions in ATLAS			
Statement	**Parameter**	**Template**	**Description**
MODELS	F.KSP	fksp ()	Energy dependent hole Peltier coefficient.
BEAM	F.OXGENERATE	oxgenerate ()	Generation rate as a function of position, time, and field (ORCHID).
DOPING	F.OXCHARGE	oxcharge ()	Position and current dependent net doping (ORCHID).
INTTRAP	F.INTDONOR	intdonor ()	Initial interface donor trap density, etc., as a function of position.
INTTRAP	F.INTACCEPTO	intacceptor ()	Initial interface acceptor trap density, etc., as a function of position.
INTTRAP	F.INTPASSIVA	intpassivated ()	Initial passivated trap density, tec., as a function of position.
INTTRAP	F.INTRATE	intrate ()	Interfere passivation rate equation versus proton flux (ORCHID).
MODELS	F.ALPHAA	bulk_absorb ()	Bulk absorption coefficient versus carrier density and photon energy.

[This page intentionally left blank]

Appendix B: Material Systems

Overview

ATLAS understands a library of materials for reference to material properties and models of various regions in the semiconductor device. These materials are chosen to represent those most commonly used by semiconductor physicists today. Users of BLAZE or BLAZE3D will have access to all of these materials. S-PISCES or DEVICE3D users will have only access to Silicon and Polysilicon.

S-PISCES is designed to maintain backward compatibility with the standalone program SPISCES2 version 5.2. In the SPISCES2 syntax, certain materials could be used in the REGION statement just by using their name as logical parameters. This syntax is still supported.

Semiconductors, Insulators and Conductors

All materials in ATLAS are strictly defined into three classes as either semiconductor materials, insulator materials or conductors. Each class of material has particular properties to which all users should be aware.

Semiconductors

All equations specified by the user's choice of models are solved in semiconductor regions. All semiconductor regions must have a band structure defined in terms of bandgap, density of states, affinity etc. The parameters used for any simulation can be echoed to the run-time output using MODELS PRINT. For complex cases with mole fraction dependent models these quantities can be seen in Tonyplot by specifying OUTPUT BAND.PARAM and saving a solution file.

Any semiconductor region that is defined as an electrode is then considered to be a conductor region. This is typical for polysilicon gate electrodes.

Insulators

In insulator materials only the Poisson and lattice heat equations are solved. Therefore for isothermal simulations, the only parameter required for an insulator is dielectric permittivity defined using MATERIAL PERM=<n>.

Materials usually considered as insulators (eg. SiO_2) can be treated as semiconductors using BLAZE, however all semiconductor parameters are then required.

Conductors

All conductor materials must be defined as electrodes. Conversely all electrode regions are defined as conductor material regions. If a file containing regions of a material known to be a conductor are read in, these regions will automatically become un-named electrodes. As noted bellow if the file contains materials that are unknown, these region will become insulators.

During electrical simulation only the electrode boundary nodes are used. Nodes that are entirely within an electrode region are not solved. Any quantities seen inside a conductor region in TONYPLOT are spurious. Only optical ray tracing and absorption for LUMINOUS and lattice heating are solved inside of conductor/electrode regions.

Unknown Materials

If a mesh file is read containing materials not in Table B-1 these will automatically become insulator regions with a relative permittivity of 3.9. All user-defined materials from ATHENA, irrespective of the material name chosen by the user, will also become such insulator materials.

ATLAS Materials

ATLAS materials are listed in Table B-1 below.

Table B-1. The ATLAS Materials

Single Element Semiconductors

Silicon[1]	Poly[2]	Germanium	Diamond

Binary Compound Semiconductors

GaAs [3]	GaP	CdSe	SnTe
SiGe	InP	CdTe	ScN
a-SiC	InSb	HgS	GaN
b-SiC	InAs	HgSe	AlN
AlP	ZnS	HgTe	InN
AlAs	ZnSe	PbS	BeTe
AlSb	ZnTe	PbSe	
GaSb	CdS	PbTe	

Ternary Compound Semiconductors

AlGaAs	GaSbP	InAlAs	GaAsP
InGaAs	GaSbAs	InAsP	HgCdTe
InGaP	InGaN	AlGaN	

Quaternary Compound Semiconductors

InGaAsP	AlGaAsP	AlGaAsSb	InAlGaN
InGaNAs	InGaNP	AlGaNAs	AlGaNP
AlInNAs	AlInNP	InAlGaAs	InAlGaP
InAlAsP			

Insulators

Vacuum	Oxide	Nitride	Si3N4
Air	SiO2	SiN	Sapphire
Ambient			

Conductors[4]

Polysilico 2	Palladium	TiW	TaSi
Aluminum	Cobalt	Copper	PaSi
Gold	Molybdenum	Tin	PtSi
Silver	Lead	Nickel	MoSi
AlSi	Iron	WSi	ZrSi
Tungsten	Tantalum	TiSi	AlSi
Titanium	AlSiTi	NiSi	Conductor
Platinum	AlSiCu	CoSi	Contact

Notes

1. The material models and parameters of Silicon are identical to those of S-PISCES version 5.2. Users should be aware that although these band parameters may be physically inaccurate compared to bulk silicon measurements, most other material parameters and models are empirically tuned using these band parameters.
2. Polysilicon is treated differently depending on how it is used. In cases where it is defined as an electrode, it is treated as a conductor. It can also be used as a semiconductor such as in a polysilicon emitter bipolars.
3. The composition of SiGe is the only binary compound that can be varied to simulate the effects of band gap variations.
4. Conductor names are only associated with electrodes. They are used for the specification of thermal conductivities and complex index of refraction and for display in `TonyPlot`.

Rules for Specifying Compound Semiconductors

The rules for specifying the order of elements for compound semiconductors are derived from the rules used by the International Union of Pure and Applied Chemistry:

1. Cations appear before anions.
2. When more than one cation is present the order progresses from the element with the largest atomic number to the element with the smallest atomic number.
3. The order of anions should be the in order of the following list: B, Si, C, Sb, As, P, N, H, Te, Se, S, At, I, Br, Cl, O, and F.
4. The composition fraction x is applied to the cation listed first.
5. The composition y is applied to the anion listed first.

To accomodate popular conventions, there are several exceptions to these rules.

- **SiGe**: The composition fraction x applies to the Ge component. SiGe is then specified as $Si_{(1-x)}Ge_{(x)}$, an exception to rule #4.
- **AlGaAs** : This is specified as $Al_{(x)}Ga_{(1-x)}As$. This is an exception to rule #2.
- **InGaAsP**: The convention $In_{(1-x)}Ga_{(x)}As_{(y)}P_{(1-y)}$ as set forth by Adachi is used. This is an exception to rule #4.

Silicon and Polysilicon

The material parameters defaults for Polysilicon are identical to those for Silicon. The following paragraphs describe some of the material parameter defaults for Silicon and Polysilicon.

Note: Within the Physics section of this manual, a complete description is given of each model. The parameter defaults listed in Chapter Three are all Silicon material defaults.

Silicon and Polysilicon Band Parameters

Table B-2. Band parameters for Silicon and Poly

Material	Eg300 eV	α	β	Nc300 per cc	Nv300 per cc	χ eV
Silicon	1.08	4.73×10^{-4}	636.0	2.8×10^{19}	1.04×10^{19}	4.17
Poly	1.08	4.73×10^{-4}	636.0	2.8×10^{19}	1.04×10^{19}	4.17

Silicon and Polysilicon Dielectric Properties

Table B-3. Static dielectric constants for Silicon and Poly

Material	Dielectric Constant
Silicon	11.8
Poly	11.8

Silicon and Polysilicon Default Mobility Parameters

The default mobility parameters for Silicon and Poly are identical in all cases. The defaults used depend on the particular mobility models in question. A full description of each mobility model and their coefficients are given in Chapter 3.

Table B-4 contains the silicon and polysilicon default values for the low field constant mobility model.

Table B-4. Lattice Mobility Model Defaults for Silicon and Poly

Material	MUN cm^2/Vs	MUP cm^2/Vs	TMUN	TMUP
Silicon	1000.0	500.0	1.5	1.5
Poly	1000.0	500.0	1.5	1.5

Table B-5 contains the silicon and polysilicon default values for the field dependent mobility model.

Table B-5. Parallel Field Dependent Mobility Model Parameters for Silicon and Poly		
Material	**BETAN**	**BETAP**
Silicon	2	1
Poly	2	1

Silicon and Polysilicon Bandgap Narrowing Parameters

The default values used in the bandgap narrowing model for SIlicon and Polysilicon are defined in Table B-6.

Table B-6. Bandgap Narrowing Parameters for Silicon and Poly			
Statement	**Parameter**	**Defaults**	**Units**
MATERIAL	BGN.E	6.92×10^{-3}	V
MATERIAL	BGN.N	1.3×10^{17}	cm^{-3}
MATERIAL	BGN.C	0.5	—

Silicon and Polysilicon Recombination Parameters

The default parameters for Schockley-Read-Hall recombination are given in Table B-7.

Table B-7. SRH Lifetime Parameter Defaults for Silicon and Poly				
Material	**TAUN0 (s)**	**TAUP0 (s)**	**NSRHN (cm^{-3})**	**NSRHP (cm^{-3})**
Silicon	1.0×10^{-7}	1.0×10^{-7}	5.0×10^{16}	5.0×10^{16}
Poly	1.0×10^{-7}	1.0×10^{-7}	5.0×10^{16}	5.0×10^{16}

The default parameters for Auger recombination are given in Table B-8;

Table B-8. Auger Coefficient Defaults for Silicon and Poly		
Material	**AUGN**	**AUGP**
Silicon	8.3×10^{-32}	1.8×10^{-31}
Poly	8.3×10^{-32}	1.8×10^{-31}

Silicon and Polysilicon Impact Ionization Coefficients

The default values for the SELB impact ionization coefficients are given in Table B-9.

Table B-9. Impact Ionization Coefficients for Silicon and Poly

Parameter	Value
EGRAN	$4.0x10^5$
BETAN	1.0
BETAP	1.0
AN1	$7.03x10^5$
AN2	$7.03x10^5$
BN1	$1.231x10^6$
BN2	$1.231x10^6$
AP1	$6.71x10^5$
AP2	$1.582x10^6$
BP1	$1.693x10^6$
BP2	$2.036x10^6$

Silicon and Polysilicon Thermal Parameters

The default values used for thermal conductivity and capacity are given in Table B-10.

Table B-10. Effective Richardson Coefficients for Silicon and Poly

Material	TCA	TCB	TCC	HCA	HCB	HCC	HCD
Silicon	0.03	$1.56x10^{-3}$	$1.65x10^{-6}$	1.97	$3.6x10^{-4}$	0.0	$-3.7x10^4$
Poly	0.03	$1.56x10^{-3}$	$1.65x10^{-6}$	1.97	$3.6x10^{-4}$	0.0	$-3.7x10^4$

Silicon And Polysilicon Effective Richardson Coefficients

Table B-11. Effective Richardson Coefficients for Silicon and Poly		
Material	**ARICHN ($A/cm^2/K^2$)**	**ARICHP ($A/cm^2/K^2$)**
Silicon	110.0	30.0
Poly	110.0	30.0

The $Al_{(x)}Ga_{(1-x)}As$ Material System

AlGaAs Recombination Parameters.

The default recombination parameters for AlGaAs are given in Table B-12.

Table B-12. Default Recombination Parameters for AlGaAs		
Parameter	**Value**	**Equation**
TAUN0	$1.0x10^{-9}$	3-213
TAUP0	$1.0x10^{-8}$	3-213
COPT	$1.5x10^{-10}$	3-226
AUGN	$5.0x10^{-30}$	3-227
AUGP	$1.0x10^{-31}$	3-227

GaAs and AlGaAs Impact Ionization Coefficients.

The default values for the SELB impact ionization coefficients used for GaAs are given in Table B-13. AlGaAs uses the same values as GaAs.

Table B-13. Impact Ionization Coefficients for GaAs	
Parameter	**Value**
EGRAN	0.0
BETAN	1.82
BETAP	1.75
EGRAN	0.0
AN1	$1.889x10^{5}$
AN2	$1.889x10^{5}$
BN1	$5.75x10^{5}$
BN2	$5.75x10^{5}$
AP1	$2.215x10^{5}$
AP2	$2.215x10^{5}$
BP1	$6.57x10^{5}$
BP2	$6.57x10^{5}$

AlGaAs Thermal Parameters.

The default thermal parameters used for AlGaAs are given in Table B-14.

Table B-14. Default Thermal Parameters for GaAs	
Parameter	**Value**
TCA	2.27
HCA	1.738

GaAs Effective Richardson Coefficients.

The default values for the effective Richardson coefficients for GaAs are 6.2875 $A/cm^2/K^2$ for electrons and 105.2 $A/cm^2/K^2$ for holes.

The $In_{(1-x)}Ga_{(x)}As_{(y)}P_{(1-y)}$ System

InGaAsP Thermal Parameters.

The default material thermal models for InGaAsP assumes lattice-matching to InP. The material density is then given by;

$$\rho = 4.791 + 0.575y.composition + 0.138y.composition$$

The specific heat for InGaAsP is given by;

$$C_p = 0.322 + 0.026y.composition - 0.008y.composition$$

The thermal resistivities of InGaAsP are linearly interpolated from Table B-15.

Table B-15. Thermal Resistivities for InGaAsP Lattice-Matched to InP

Composition Fraction y	Thermal Resistivity (deg(cm/w)
0.0	1.47
0.1	7.05
0.2	11.84
0.3	15.83
0.4	19.02
0.5	21.40
0.6	22.96
0.7	23.71
0.8	23.63
0.9	22.71
1.0	20.95

The default thermal properties of the binary compounds in the InGaAsP system are given in Table B-16.

Table B-16. Default Thermal Properties of InP InAs GaP and GaAs

Material	Thermal Capacity (J/cm3)	Thermal Resistivity (deg(cm/W)
InP	1.543	1.47
InAs	1.994	3.70
GaP	1.292	1.30
GaAs	1.738	2.27

The default thermal properties for the terniary compounds in the InGaAsP system: $In_{(1-x)}Ga_{(x)}As$, $In_{(1-x)}Ga_{(x)}P$, $InAs_{(y)}P_{(1-y)}$, and $GaAs_{(y)}P_{(1-y)}$ are given, as a function of composition fraction, by linear interpolations from these binary compounds.

Silicon Carbide (SiC)

SiC Impact Ionisation Parameters

The default values for the SELB impact ionization coefficients used for SiC are given in Table B-17.

Table B-17. Impact Ionization Coefficients for SiC

Parameter	Value
EGRAN	0.0
BETAN	1.0
BETAP	1.0
AN1	$1.66x10^6$
AN2	$1.66x10^6$
BN1	$1.273x10^7$
BN2	$1.273x10^7$
AP1	$5.18x10^6$
AP2	$5.18x10^6$
BP1	$1.4x10^7$
BP2	$1.4x10^7$

SiC Thermal Parameters.

The default thermal parameters used for both 6H and 4H-SiC are shown in Table B-18.

Table B-18. Default Thermal Parameters for SiC

Parameter	Value	
	4H-SiC	6H-SiC
TCA	0.204	0.385
HCA	0	0

Miscellaneous Semiconductors

The remainder of the semiconductors available have defined default parameter values to various degrees of completeness. The following sections describe those parameter defaults as they exist. Since many of the material parameters are not available at this time, it is recommended that care be taken in using these materials. It is important to make sure that the proper values are used

Note: The syntax `MODEL PRINT` can be used to echo the parameters used to the run-time output.

Miscellaneous Semiconductor Band Parameters

Table B-19. Band Parameters for Miscellaneous Semiconductors

Material	Eg(0)eV	Eg(300)eV	α	β	m_c	m_v	χeV
Silicon							
Poly-silicon							
Ge	0.7437		$4.77x10^{-4}$	235.0	0.2225	0.2915	4.0
Diamond		5.45	$4.77x10^{-4}$	0.0	(a)	(b)	7.2
6H-SiC	2.9	2.9	0.0	0.0	0.454	0.33	
4H-SiC	2.2	2.2	0.0	0.0	0.41	0.165	
A1P	2.43	2.43	0.0	0.0			
A1As	2.16	2.16	0.0	0.0			
A1Sb	1.6		$2.69x10^{-4}$	2.788	(c)	0.4	
GaSb	0.81		$3.329x10^{-4}$	-27.6622	(c)	0.24	3.65
InSb	0.235		$2.817x10^{-4}$	90.0003	0.014	0.4	4.06
ZnS	3.8	3.8	0.0	0.0	0.4		4.59
ZnSe	2.58	2.58	0.0	0.0	0.1	0.6	
ZnTe	2.28		0.0	0.0	0.1	0.6	4.09
Cds	2.53	2.53	0.0	0.0	0.21	0.8	3.5
CdSe	1.74	1.74	0.0	0.0	0.13	0.45	4.5
CdTe	1.5	1.5	0.0	0.0	0.14	0.37	
HgS	2.5	2.5	0.0	0.0			4.28
HgSe							
HgTe							

Table B-19. Band Parameters for Miscellaneous Semiconductors							
Material	**Eg(0)eV**	**Eg(300)eV**	α	β	m_c	m_v	χeV
PbS	0.37	0.37	0.0	0.0	0.25	0.25	
PbSe	0.26	0.26	0.0	0.0	0.33	0.34	
PbTe	0.29	0.29	0.0	0.0	0.17	0.20	4.6
SnTe	0.18	0.18	0.0	0.0			
ScN	2.15	2.15	0.0	0.0			
GaN	3.45	3.45	0.0	0.0	0.172	0.259	
A1N	6.28	6.28	0.0	0.0	0.314	0.417	
InN	1.89	1.89	0.0	0.0	0.11	0.17	
BeTe	2.57	2.57	0.0	0.0			

Notes

(a). $Nc300 = 5.0x10^{18}$

(b). $Nv300 = 1.8x10^{19}$

(c). $m_c(X) = 0.39$

$m_c(G) = 0.09$

$Nc = Nc(X) + Nc(G)$

(d). $m_c(G) = 0.047$

$m_c(L) = 0.36$

$Nc = Nc(G) + Nc(L)$

Miscellaneous Semiconductor Dielectric Properties

Table B-20. Static Dielectric Constants for Miscellaneous Semiconductors	
Material	**Dielectric Constant**
Ge	16.0
Diamond	5.5
6H-SiC(a)	9.66
4H-SiC(b)	9.72
AlP	9.8
AlAs	12.0
AlSb	11.0

Table B-20. Static Dielectric Constants for Miscellaneous Semiconductors	
Material	**Dielectric Constant**
GaSb	15.7
InSb	18.0
ZnS	8.3
ZnSe	8.1
CdS	8.9
CdSe	10.6
CdTe	10.9
HgS	
HgSe	25.0
HgTe	20.
PbS	170.0
PbSe	250.0
PbTe	412.0
SnTe	
ScN	
GaN	9.5
AlN	9.14
InN	19.6
BeTe	

Miscellaneous Semiconductor Mobility Properties

Table B-21. Mobility Parameters for Miscellaneous Semiconductors				
Material	**MUNO (cm2/Vs)**	**MUPO (cm2/Vs)**	**VSATN(cm/s)**	**VSAT(cmcm/s)**
Ge	3900.0(a)	1900.0(b)		
Diamond	500.0	300.0	$2.0x10^7$	
SiC(a)	330.0	300.0	$2.0x10^7$	
SiC(b)	1000.0	50.0	$2.0x10^7$	
AlP	80.0			

Table B-21. Mobility Parameters for Miscellaneous Semiconductors				
Material	**MUNO (cm2/Vs)**	**MUPO (cm2/Vs)**	**VSATN(cm/s)**	**VSAT(cmcm/s)**
AlAs	1000.0	100.0		
AlSb	200.0	550.0		
GaSb	4000.0	1400.0		
InSb	7800.0	750.0		
ZnS	165.0	5.0		
ZnSe	100.0	16		
CdS	340.0	50.0		
CdSe	800.0			
CdTe	1050.0	100.0		
HgS				
HgSe	5500.0			
HgTe	22000.0	100.0		
PbS	600.0	700.0		
PbSe	1020.0	930.0		
PbTe	6000.0	4000.0		
SnTe				
ScN				
GaN	400.0	8.0	$2.0x10^7$	
AlN		14.0		
InN	3000.0			
BeTe				

Notes

(a) Uses Equation B-4 with TMUN=1.66.

(b) Uses Equation B-4 with TMUP = 2.33.

Insulators

The default material parameters for insulator materials are given in the following sections. As noted in the “Semiconductors, Insulators and Conductors” section the only parameter required for electrical simulation in insulator materials is the the dielectric constant .Thermal and optical properties are required in GIGA and LUMINOUS respectively.

Insulator Dielectric Constants

Table B-22. Default Static Dielectric Constants of Insulators	
Material	**Dielectric Constant**
Vacuum	1.0
Air	1.0
Ambient	1.0
Oxide	3.9
Si02	3.9
Nitride	7.5
SiN	7.5
Si3N4	7.55
Sapphire	12.0

Insulator Thermal Properties

Table B-23. Default Thermal Parameters for Insulators			
Material	**Thermal Capacity (J/cm^3)**	**Thermal Conductivity(deg(cm/W)**	**Reference**
Vacuum	0.0	0.0	
Air	1.0	0.026	7
Ambient	1.0	0.026	7
Oxide	3.066	0.014	4
Si02	3.066	0.014	4
Nitride	0.585	0.185	4

Table B-23. Default Thermal Parameters for Insulators			
Material	**Thermal Capacity (J/cm^3)**	**Thermal Conductivity(deg(cm/W)**	**Reference**
SiN	0.585	0.185	4
Si3N4	0.585	0.185	4
Sap-phire			

Optical Properties

The default values for complex index of refraction in LUMINOUS are interpolated from tables from the "Handbook of Optical Constants," first and second editions. Rather than print the tables here, the ranges of optical wavelengths for each material are listed in Table B-24.

Table B-24. Wavelength Ranges for Default Complex Index of Refraction

Material	Temperature(K)	Composition Fraction	Wavelengths (microns)
Silicon	300	NA	0.0103-2.0
AlAs	300	NA	0.2213 - 50.0
GaAs	300	NA	0.0 - 0.9814
InSb	300	NA	0.2296 - 6.5
InP	300	NA	0.1689 - 0.975
Poly	300	NA	0.1181 - 18.33
SiO2	300	NA	0.1145 - 1.7614

Note: The parameter INDEX.CHECK can be added to the SOLVE statement to list the values of real and imaginary index being used in each solution.

User Defined Materials

The current version of ATLAS does not directly support user defined materials. A simple workaround can be done using the already existing user specifications. This workaround is based on the use of an already existing material name and modifying the material parameters as appropriate.

In ATLAS material names are defined to give the user a reasonable set of default material parameters. Any of these defaults can be overriden using the MATERIAL, IMPACT, MODEL, and MOBILITY statements. The key to defining new materials is choosing a material name that is defined in ATLAS, then modifying the material parameters of that material to match the user material. Here it is best to choose a material that has default parameter values that might best match the user material, while being sure to choose a material that is not already in the user device. Next the user must associate this material name with the device regions where the new material is present. This is done by either specifying the chosen material name on the appropriate REGION statements (when the device is defined in the ATLAS syntax) or choosing the material name from the materials menu when defining the region in DEVEDIT.

Next, the user should modify the material statements using MATERIAL, IMPACT, MOBILITY, and MODEL statements. When doing this the MATERIAL parameter of the given statement should be assigned to the chosen material name.

For materials with variations in composition fraction, the user should choose a defined material with X and/or Y composition fractions (i.e., a terniary or quaterniary material). The user may also find it convenient to use C interpreter functions to define the material parameters as a function of composition. The C interpreter functions that are useful for this approach are: F.MUNSAT, F.MUPSAT, F.BANDCOMP, F.VSATN, F.VSATP, F.RECOMB, F.INDEX, F.BGN, F.CONMUN, F.CONMUP, F.COPT, F.TAUN, F.TAUP, F.GAUN, and F.GAUP.

In defining new materials there exists a minimum set of parameters that should be defined. This set includes bandgap (EG300), electron and hole density of states (NC300 and NV300), dielectric permitivity (PERMITIVITY), and electron and hole mobilities (MUN and MUP). For bipolar devices certain recombination parameters should also be defined such as: lifetimes (TAUN and TAUP), radiative recombination rates (COPT), and Auger coefficients (AUGN and AUGP). For devices with variations in material composition certain band-edge alignment parameters should also be defined: either electron affinity (AFFINITY) or edge alignment (ALIGN). If impact ionization is considered the impact ionization coefficients should also be defined.

As an example, consider the case where the user is simulating a device with an AlInGaP region. Consulting table B-1, we see that this material system is not defined in ATLAS. We then choose a materal that is defined in ATLAS which has default material parameters that best approximate the material parameters of the new material. In this case, we choose InGaAsP since, at least for example purposes, we feel that this material is closest to the AlInGaP. Next, we must specify InGaAsP as the material of the region(s) that is/are composed of AlInGaP. This can be done either on the REGION statement if the structure is defined in ATLAS syntax or from the material menu when the region is defined in DEVEDIT.

Supposing that we are satisfied with the default values of the parameters from the "minimum set" discussed above, and that we are principally concerned with the recombination and heat flow parameters defaults, the following section of the input deck illustrates how these parameter defaults may be modified:

```
# new material AlInGaP
MATERIAL MATERIAL=InGaAsP
# SRH
MATERIAL  MATERIAL=InGaAsP TAUN0=1.1e-9 TAUP0=2.3e-8
# Auger
MATERIAL MATERIAL=InGaAsP AUGN=5.8e-30 AUGP=1.1e-31
# Optical
material material=InGaAsP COPT=1.7e-30
```

```
# Thermoconductivity
MATERIAL MATERIAL=InGaAsP TC.A=2.49
# Heat capacity
MATERIAL  MATERIAL=InGaAsP HC.A=1.9
```

Appendix C: Hints and Tips

This appendix is a collection of answers to commonly asked questions about the operation of ATLAS. This information has been previously published in articles in The Simulation Standard™, Silvaco's trade publication. The original articles can be viewed at Silvaco's home page at http://silvaco.com/simstd.

Question:

Can ATLAS handle photogeneration from non-normally incident light? What syntax is used to determine the correct photogeneration rate?

Answer:

The models with LUMINOUS allow simulation of photogeneration within ATLAS. Arbitrary light sources are available within Luminous using the BEAM statement. The program uses geometric ray tracing to determine the path of a light beam includingrefraction and reflection of non-normally incident light. It then applies models for the absorption of the light to determine the photogeneration rate.

To use luminous the use should first define a light source using the BEAM statement and then choose the light intensity using a SOLVE statement. The BEAM statement can be explained by a simple example. The following syntax generates the ray trace in Figure C-1.

```
BEAM NUM=1 X.ORIGIN=3.0 Y.ORIGIN=-5.0 ANGLE=60.0 \
MIN.W=-1 MAX.W=1 WAVEL=0.6 REFLECT=2
```

The parameter NUM sets the beam number. LUMINOUS can applications up to 10 independent beams. X.ORIGIN and Y.ORIGIN define the initial starting point for the light beam. This must be outside the device coordinates and for non-normal beams it is important to keep this point well away from the device. The ANGLE parameter determines the direction of the light beam relative to the x-axis. ANGLE=90 gives normal incidence from the top of the device. The light is defined as coming from a line perpendicular to the direction set by ANGLE and passing through (X.ORIGIN, Y.ORIGIN). MAX.W and MIN.W set a window along this line through which the light passes. The default for these parameters is +/- infinity.

The wavelength of the light beam is defined in microns using the WAVEL parameter. It is also possible to specify spectral sources of light by replacing WAVEL with POWER.FILE={filename}, where the {file} is a UNIX text file defining the spectrum in terms of an XY list of wavelength vs intensity.

LUMINOUS automatically chooses the number of rays needed to resolve the geometry. In the case of Figure C-1, only one initial ray is needed. The number of rays used by LUMINOUS is purely a function of the geometry and is not related to optical intensity, photogeneration rate, or MIN.W and MAX.W. The REFLECT parameter is used to limit the number of reflections the light beam is allowed. In ATLAS Version 3.0.0.R an alternative parameter MIN.POWER is used to set a relative intensity below which no more rays are traced.

The user has a choice in LUMINOUS as to whether the rays should reflect from the sidewalls and bottom of the device structure. If the simulation is of a partial wafer (such as is typical for CCD simulation) the light should not reflect. This is the default shown in Figure C-1. If the simulation is of a complete device, such as is typical for solar cells, the light should reflect. The parameter BACK.REFL is used to enable back and side reflection. The result of adding this to the previous syntax is shown in Figure C-2. In this figure the limit set by REFLECT=2 is also clear as each ray reflects only two times.

The ray trace is only done once a SOLVE statement is used to turn on the light beam. For example, the syntax `SOLVE B1=0.5` sets the power of beam #1 to 0.5W/cm^2. DC and transient ramps of light intensity can also be performed.

The refraction and reflection are determined by the real portion of the refractive index for each material. The imaginary portion on the refractive index controls the absorption of the light. Wavelength dependent defaults exist in ATLAS for common materials, but can be defined by the user as follows:

```
MATERIAL MATERIAL=Silicon REAL.INDEX={value} IMAG.INDEX={value}
```

From the ray trace information and the imaginary refractive indices, the photogeneration rate is calculated at each node in the structure. An integration method is used to ensure the charge generated is independent of the mesh density. The photogeneration rate from the ray trace in Figure C-2 is shown in Figure C-3. The electrons and holes generated are included in electrical simulations in ATLAS to determine collection efficiency, spectral response and other device parameters.

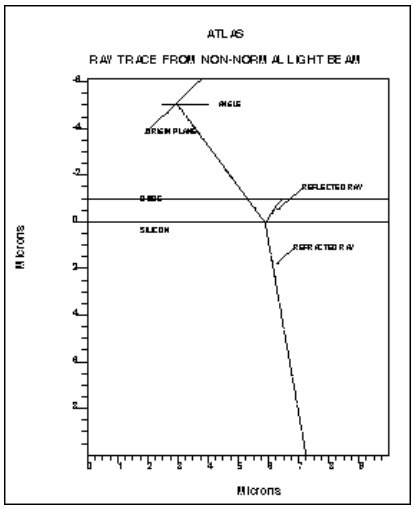


Figure C-1: Simple ray trace in LUMINOUS

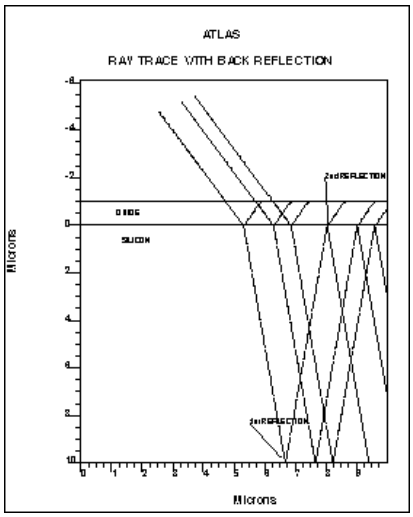


Figure C-2: Addition of back and sidewall reflection to Figure C-1

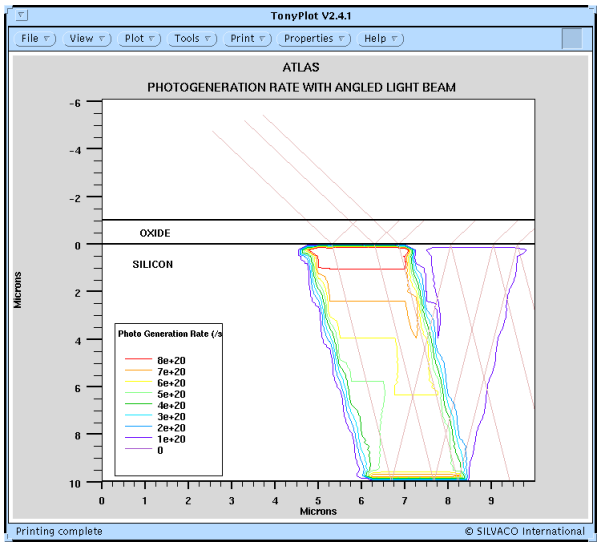


Figure C-3: Photogeneration contours based on ray trace in Figure C-2

Question:

What choices of numerical methods are available in ATLAS? When should each type of method be used?

Answer:

The latest release of ATLAS features more choices of numerical methods for users. It also has a new more logical syntax that clears up some of the previously confusing issues with the choice of numerical method for ATLAS solutions.

ATLAS has the ability to solve up to six equations on the simulation mesh. These are the Poisson equation, two carrier continuity equations, two carrier energy balance equations, and the lattice heat flow equation. The choice of numerical technique in solving these equations can strongly affect both the convergence and CPU time required to complete a simulation run.

In general equations can either be solved in a coupled manner with all equations solved at once or a decoupled manner with a subset of the equation solved whilst others are held constant. The coupled solutions are best when the interactions between the equations is strong (i.e., high current producing significant local heating). However they require a good initial guess to the solution variables for reliable convergence. Unless special techniques, such as projection, are used for calculating initial guesses this would mean that the voltage step size during a bias ramp might have to be rather small.

Decoupled methods can have advantages when the interaction between the equations is small (typically low current and voltage levels). They do not require such good initial guesses for convergence. They tend to either not converge or take excessive CPU time once the interactions become stronger.

ATLAS uses the METHOD only statement to define numerical methods. The older SYMBOLIC statement is no longer required, although existing input files will run as before. To select the decoupled method for two carriers use:

```
method gummel carriers=2
```

To select the coupled method for two carriers use:

```
method newton carriers=2
```

This is the default. In fact users do not need to specify a METHOD statement at all for this case. For the majority of isothermal drift-diffusion simulations this choice of numerics is the recommended one.

ATLAS has the ability to switch automatically between different numerical methods. The syntax

```
method gummel newton
```

would start each bias point using a decoupled method and then switch to a coupled solution. This technique is extremely useful when the initial guess is not good. Typically this happens for devices with floating body effects (i.e., SOI).

For more complex simulation with energy balance or lattice heating other techniques are also available in ATLAS. A mixed technique where the poisson and continuity equations are solved coupled, and then the other equations are decoupled can be applied.

The syntax for this is:

```
method block
```

Typically this mixed technique is quicker and more robust at low lattice and carrier temperatures, whereas the fully coupled technique is better for high lattice and carrier temperatures. The mixed method can be combined very effectively with the fully coupled technique to provide improved speed and convergence for all ranges using:

```
method block newton
```

As an example of this Figure C-4 shows the CPU time taken for individual biasing points for a non-isothermal energy balance simulation of second breakdown of a sub-micron MOSFET. The device was constructed using ATHENA and the mesh contains a low number of obtuse triangles (not zero). The graph clearly shows that at initially the coupled method takes much longer to converge. This is because the initial guess is not good until two bias points are solved so that projection can be used. As the voltage is increased the block method has increasing convergence difficulty whereas the time taken for the newton method is flat. The time for the mixed method has the advantages of block at lower currents, but without the severe increases of block.

The ideal solution for most problems is the use of the mixed "block newton" method initially switching to the fully coupled "newton" method as the simulation proceeds. Of course the cross-over point, which is at x=2.5 in Figure C-4, varies from case to case. To avoid excessive user-interaction the "block newton" could be used throughout the simulation without a very excessive hit in terms of CPU time.

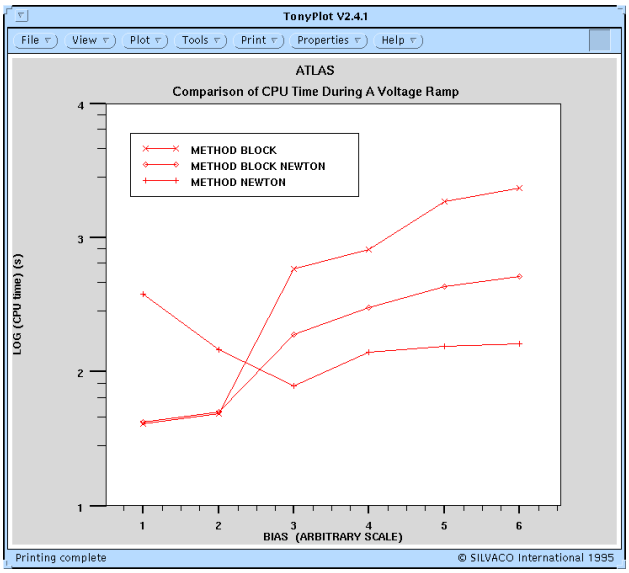


Figure C-4: Comparison of CPU time showing advantages of decoupled methods at low current and coupled method at high currents

Question:

Can the workfunction of the MOS polysilicon gate contact be calculated by ATLAS based on the doping? Can poly depletion effects be simulated in ATLAS?

Answer:

The polysilicon gate contact in MOS devices can be simulated in two distinct ways using ATLAS. These correspond to treating the polysilicon region as:

- A metal-like equipotential region with a specified workfunction.
- A semiconductor region with a potential defined by the doping level.

Most commonly the former approach is adopted. The polysilicon region acting as the gate is defined as an electrode in ATHENA.

The `electrode` statement is used with the X and Y parameters acting as crosshairs to target a particular region of the structure. The whole region, irrespective of shape, is then defined as an electrode.

```
ELECTRODE NAME=gate X={x value} Y={y value}
```

A region defined this way is now treated as equipotential in ATLAS. The potential of this region will be defined by the VGATE parameter of the SOLVE statement. Hence poly depletion cannot be modeled in gate contacts defined this way. The workfunction of this region must be set by the user. For example, a heavily n doped polysilicon contact can be defined by either of the two following statements:

```
CONTACT NAME=gate N.POLY
CONTACT NAME=gate WORK=4.17
```

The second approach of treating the polysilicon gate region as a semiconductor is achieved by placing a contact on the top of the gate. In ATHENA this is done be depositing a metal (or silicide) layer on top of the polysilicon. The ELECTRODE statement is then used to define this metal region as the gate electrode. In ATLAS a workfunction for the gate should not be specified on the CONTACT statement as this would give an undesirable workfunction difference between the metal and polysilicon. the potential on the metal region is defined by the VGATE parameter of the SOLVE statement. The potential within the polysilicon gate region will depend on the doping level of the polysilicon. In non-degenerately doped polysilicon a voltage drop is seen across this region from top to bottom. The workfunction difference between the gate and the substrate can be derived from a potential profile through the channel region.

It is also possible for the polysilicon to be depleted starting at the gate oxide interface. Figure C-5 shows a comparison of high frequency CV curves between a MOS device with a uniform degenerately doped poly gate typical when tube doping is used and a lighter, non-uniformly doped gate typical when source/drain implants are used to dope the polysilicon. In the accumulation region the poly begins to deplete leading to an effectively thicker gate dielectric. This effect is illustrated in Figure C-6. The amount of poly depletion observed is dependent on the doping level. Accurate polysilicon diffusion models are available in ATHENA to simulate the doping. In addition the SILICIDE module allows simulation of the dopant redistribution during gate silicidation. Silicides can typically reduce the effective gate doping making poly depletion more likely.

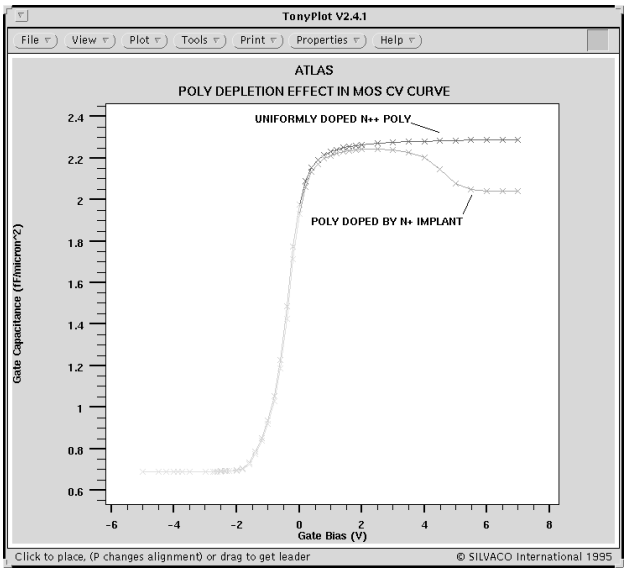


Figure C-5: High frequency CV curve showing poly depletion effects at positive Vgs.

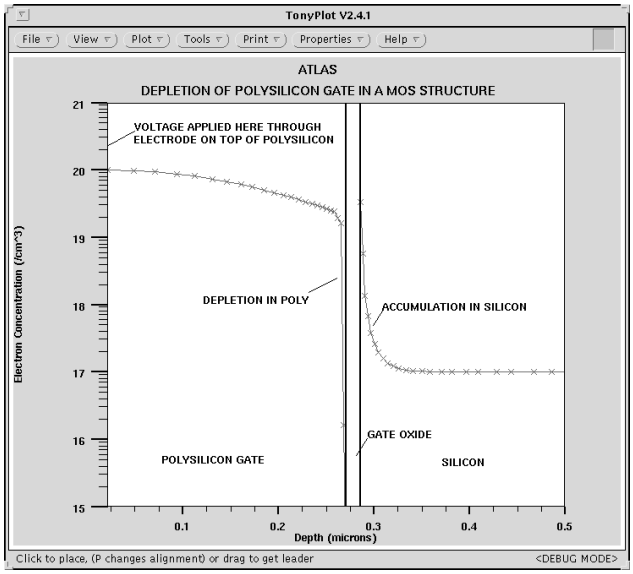


Figure C-6: Figure 2 Electron concentration profile of an NMOS transistor poly depletion occurs at the poly/gate oxide interface.

Question:

How can I remesh my process simulation result for device simulation?

Answer:

The structure editing and griding tool DEVEDIT provides an effective way to remesh structures between process and device simulation.

DEVEDIT is able to read and write files to both ATHENA and ATLAS. In addition to the graphical user interface, DEVEDIT can operate in batch mode under DECKBUILD. The batch mode DEVEDIT features a powerful syntax that can be applied to remesh structures read directly from ATHENA. DEVEDIT employs a heirachical method for remeshing existing structures. Users specify mesh parameters at each stage. Initially a base mesh is applied with the command:

```
BASE.MESH HEIGHT=<h1> WIDTH=<w1>
```

This mesh of h1 by w1 microns provides the coarsest mesh in the structure. On top of this base level mesh DEVEDIT determines which points must be added to ensure the geometry of all regions is preserved. Optional boundary conditioning to smooth region boundaries can be applied using the BOUND.COND statement.

Mesh constraints can be applied to arbitrary boxes with the device using the syntax:

```
constr.mesh x1=<n> x2=<n> y1=<n> y2=<n>\
max.height=<h2> max.width=<w2>
```

This sets the maximum mesh size to h2 by w2 microns the box with diagonal from (x1,y1) to (x2,y2). Using the constraint boxes in critical areas of the device is the most effective way to use DEVEDIT. In MOSFETS the constraint boxes can be applied to the silicon region under the gate. Typically vertical grid spacings of 5 are required for accurate simulation of channel conduction in sub-micron MOSFETs.

Use of multiple constraint boxes can be applied. For MOSFET breakdown constraint boxes can be applied to the drain/gate overlap area.

In addition to constraint boxes, DEVEDIT can refine on quantities such as individual doping species, net doping, potential. The IMP.REFINE statement allows users to select the critical value of a quantity for refinement and an associated minimum grid spacing.

This type of refinement is useful when key effects are located very close to junctions. For example the emitter-base junction in bipolar transistors.

The final level of grid refinement are "manual refine boxes" defined using:

```
Refine Mode={Both|x|y} P1=x1,y1 P2=x2,y2
```

These are boxes with a diagonal from (x1,y1) to (x2,y2), inside which the grid spacing is halved in the given direction. These can be used as a final customization to the mesh. In general, however, the constraint boxes described above are easier to use.

As a demonstration of the effectiveness of regriding using devedit Figures C-7 and C-8 show meshes created from the same process simulation structure. Figures C-9 and C-10 show the Id/Vds and breakdown characteristics respectively of the device for each mesh. The mesh at the silicon surface is identical in both cases to a depth of 100A. However the mesh below this is much coarser in Figure C-8.

However the device results show no significant change despite the reduction in the node count by a factor of two.

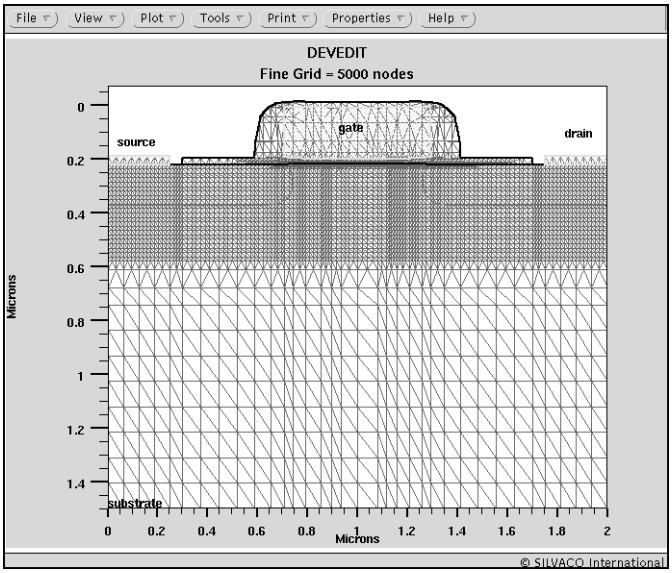


Figure C-7: Fine Grid - 5000 nodes

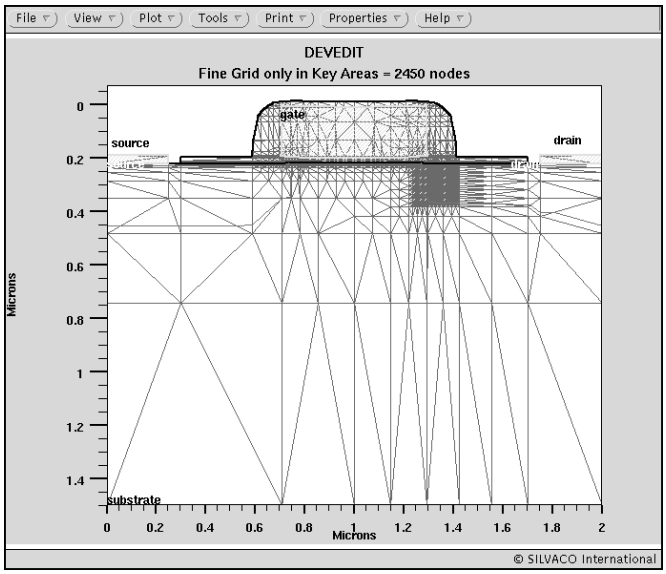


Figure C-8: Fine Grid only in Key Areas - 2450 nodes

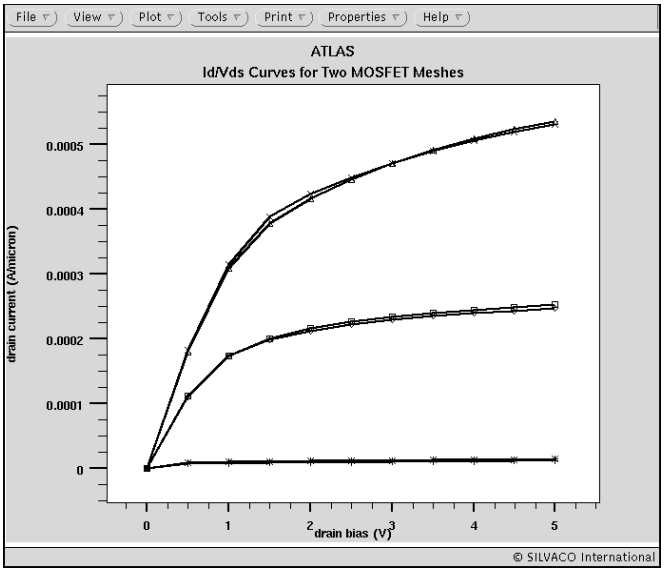


Figure C-9: Id-Vds Curves for two MOSFET Meshes

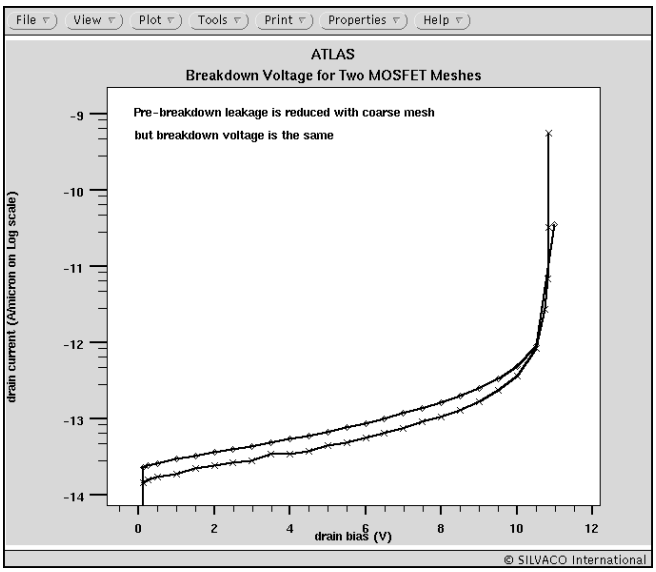


Figure C-10: Breakdown Voltage for Two MOSFET Meshes

Demonstration of optimized griding using constraint boxes in DEVEDIT. Both very fine and optimized mesh produces equivalent results.

Question:

How can solution quantities such as Electric Field be saved for plotting against applied bias?

Answer:

There are two types of output files saved by ATLAS:

- Solution files contain physical quantities mapped to the simulation grid. One solution file is saved per bias point.
- Log files which traditionally have saved the terminal characteristics for all bias points.

A new feature of ATLAS **4.0** is the addition of a capability to save physical quantities at user-specified locations within the device grid to the log files. A new statement PROBE is used to specify the quantity to be saved and the location. For example, to save the electron concentration rate at a location [1,0.1] the syntax is:

```
PROBE NAME=my_e_conc N.CONC X=1.0 Y=0.5
```

The label specified by the NAME parameter is used to label the saved quantity in TONYPLOT and subsequent EXTRACT statements.

For vector quantities the PROBE statement also requires a direction to be given using the DIR parameter. This is specified as an angle in degrees with the X axis as DIR=0. To find the electric field across an oxide layer the syntax is:

```
PROBE X=1.2 Y=0.2 FIELD DIR=90 \ NAME=oxide_field
```

Figure C-11 shows the resultant plot of electric field in a MOSFET gate oxide during a transient ESD pulse. This result shows the probability of oxide breakdown during the ESD stress, without the need to examine many separate solution files.

Another advantage of the probe for vector quantities is that it reads the values directly from the simulator at the closest location to the specified XY coordinates. This avoids many issues of interpolation and averaging of vector quantities onto the simulation grid.

If two physical quantities are probed at the same location it is possible to plot them against each other to examine model settings. For example, impact ionization rate or mobility versus electric field. Figure C-12 shows a plot of channel electron mobility in a submicron NMOS transistor versus the transverse electric field from the gate.

All of the primary solution quantities can be probed. A full list is given in the manual under the PROBE statement. In addition to values at point locations the PROBE statement also supports MIN and MAX parameters to find the minimum and maximum of a given quantity.

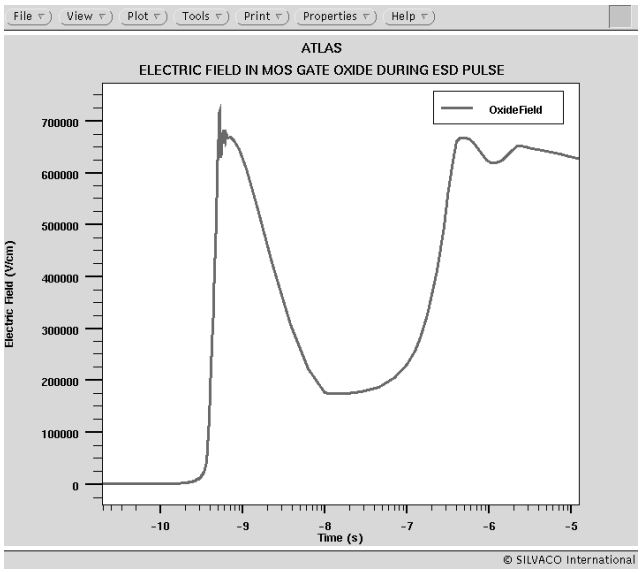


Figure C-11: Electric field in MOS gate oxide during a high current pulse on the drain.

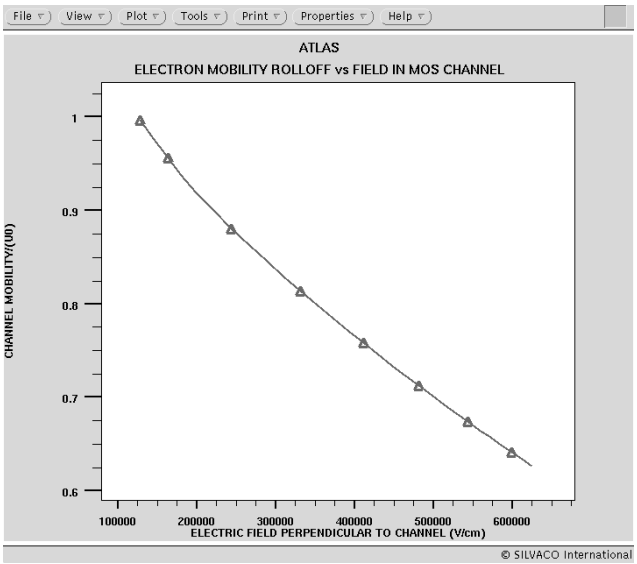


Figure C-12: Mobility (normalized) rolls off as a high gate electric field is applied

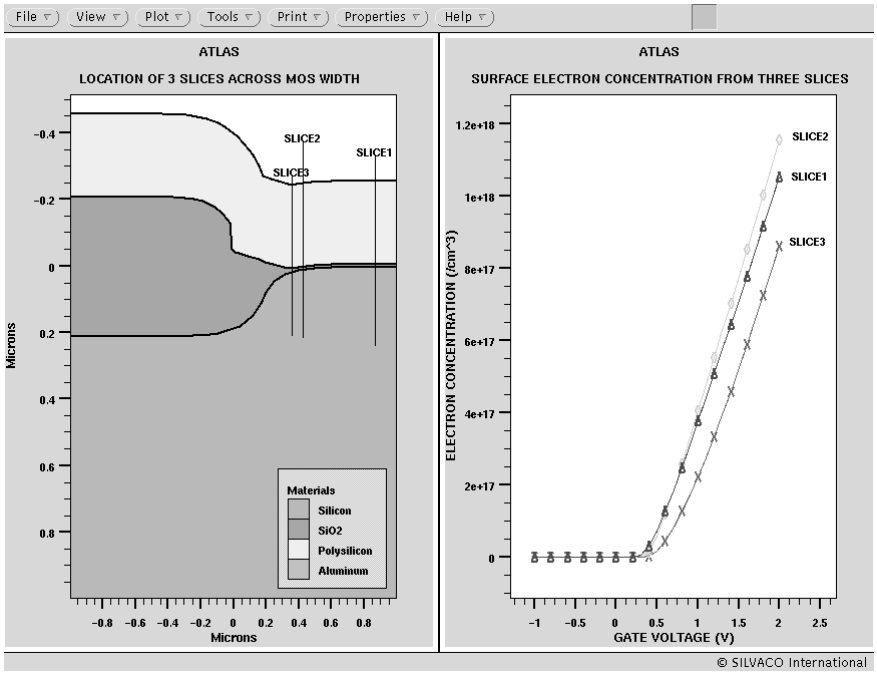


Figure C-13: Using a PROBE of electron concentration allows a study of MOS width effect using 2D simulation. An enhanced electron concentration is seen along slice 2.

Question:

What are the options for generating 3D structures for ATLAS device simulation?

Answer:

Currently there are three options for generating 3D device structures. In all cases the prismatic meshing of ATLAS/DEVICE3D permits arbitrary shaped device regions in 2 dimensions (typically X and Y) and rectangular regions in the other dimension (typically Z).

- Definition though the ATLAS syntax.

This limits the user to defining box shaped regions. Region definition is though statements such as:

```
region num=1 silicon x.min=0  \
x.max=1 y.min=0 y.max=1 z.min=0
\ z.max=1
```

Mesh generation is handled though the `Z.MESH` statement which is analogous to the `X.MESH` and `Y.MESH` statement used in 2D ATLAS simulations. Electrodes and doping can be defined using the same syntax as 2D ATLAS but with `Z.MIN` and `Z.MAX` parameters to control the Z extent. Doping profiles can be read from the formats supported in 2D ATLAS: `ssuprem3, athena, ascii`.

- Use DEVEDIT to create a 3D mesh structure.

DEVEDIT3D is a 3D structure generation and meshing tool used to generate the mesh, regions, electrodes and doping used in ATLAS/DEVEDIT. It allows users to draw regions by hand in 2D and project them into the third direction.

DEVEDIT3D contains all the sophisticated meshing options available to 2D DEVEDIT. These include: mesh constraints by region, mesh constraints by a user defined box, refinement on doping and other

quantities, mouse controlled refine and unrefine boxes. DEVEDIT3D has both interactive and batch mode operation. This is the recommended approach for 3D power device simulation.

- Use ATHENA and DEVEDIT3D to extend a 2D process simulation to 3D.

A 2D process simulation from ATHENA can be extruded to form a 3D structure using DEVEDIT3D. In this mode DEVEDIT3D can be used to add extra regions and doping if required. This mode us commonly used for modeling MOS width effects in 3D (see Figure C-14). An ATHENA simulation of the isolation bird's beak and field stop implant is performed.

The 2D structure is loaded into DEVEDIT3D and extended in Z direction. The polysilicon gate is truncated in Z and aluminum source/drain contacts are added. The source/drain doping profiles can be read from analytical functions or other process simulation results. A worked example, named mos2ex04.in, is supplied with the Fall 96 release CDROM.

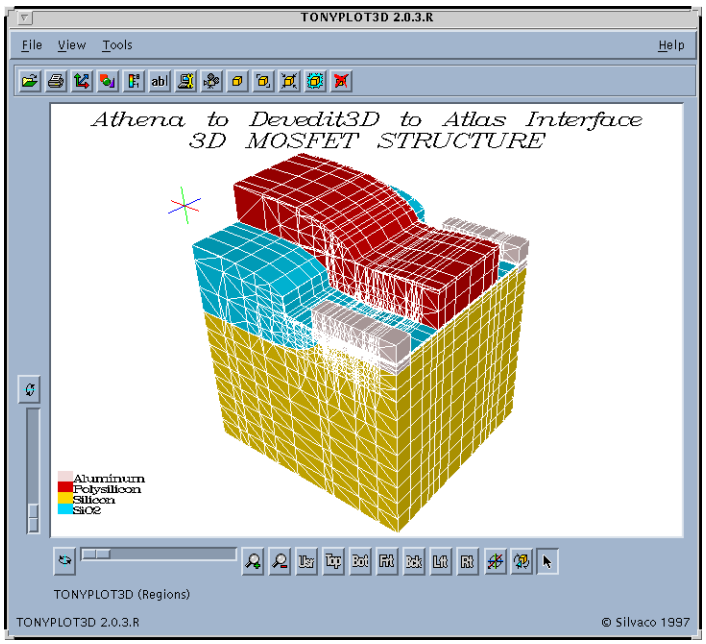


Figure C-14: 3D device simulation of MOS width effect can be performed on structures created ATHENA.

For devices with non-rectangular regions in the XZ plane such as pillar MOSFETs or cylindrically designed power structures DEVEDIT3D can also be used by drawing in the XZ plane and projecting into the Y direction.

The future release of Silvaco's 3D process/device simulator **ODIN** will provide a fourth option for 3D simulation problems. ODIN is based on tetrahedral mesh elements and will overcome the mesh restrictions described above.

[This page intentionally left blank]

Appendix D: ATLAS Version History

Version History

Version 5.2.0.R

1. **Introduced LUMINOUS3D**
   - **LUMINOUS3D** provides a similar functionality to 2D LUMINOUS. This allows simulation of interaction between arbitrary optical sources and semiconductor devices in three dimensions.
2. **Smartlib**
   - Implemented the SmartSpice Compact Device Model Library (Smartlib). These models replaces the old MIXEDMODE Diode, MOSFET and BJT Models. Additional device models such as TFT and SOI can now be used. Also, additional devices such as JFETS and MESFETs can now be used.
3. **Implemented a New and Improved C-Interpreter**
   - Implemented a new C-Intrepreter. This completely replaces the old version of the C-Interpreter. The new C-Interpreter is faster than the old version and supports more C functions. The new C-Interpreter conforms more to the ANSI standard.
4. **Hetereojunction Thermionic Emission and Field Emission**
   - Introduced a new implementation of thermionic emission models for hetero-interfaces. This implementation supercedes the previous one, which is no longer supported.
   - Also introduced a new heterojunction tunneling (thermionic field emission) model that is compatible with the new thermionic emission model.
5. **Cyclic Biasing**
   - Implemented cyclic biasing model to simplify prediction of steady state conditions in cyclically biased devices exhibiting long time constants.
6. **Miscellaneous Improvements**
   - Added the following new material names: InGaN, AlGaN, InAlGaN, inganas, InGaNAs, inGaNP, AlGaNAs, AlGaNP, A1InNAs, AlInNP, InA1GaAs, InA1GaP and InA1AsP.
   - Added the following new gain output parameters. ATLAS will now ouput the maximum available gain (Bma) as well as the maximum stable gain (Gms).
   - Added capability to add sheet charge along region interfaces for simulation of piezoelectric strain effects.
   - Added temperature dependence to the Shirahata mobility model.
   - Added capability for multiple trapazodial/square transient pulses.
   - Added a more general parameterized temperature-dependent model for saturation velocity for field-dependent mobility.
   - Changed the calculation of effective mass used in calculation of thermal velocity for traps in noo-silicon materials. The new approach calculates mass, based on the material density of states.

Version 5.0.0.R

1. Introduced several new 3D products

a. Giga3D.

- **Giga3D** contains most of the functionality of the 2D Giga, but works with 3D products. This allows modeling of heatflow and self-heating effects in 3D devices. The only functionality not supported in this version of **Giga3D** that is supported in 2D is the BLOCK method.

b. Blaze3D

- This version accounts for spatial variations in band gap due to spatial variations in material composition in 3D. This version supports all the same models that are supported in Blaze 2D with the exception of thermionic emission at hetereojunctions and energy transport. This version also does not support compositional variation in the z direction.

c. MixedMode3D

- This improvement allows simulation of 3D devices embedded in lumped element circuits. **MixedMode3D** contains all the functionality of 2D MixedMode simulator.

d. TFT3D

- This model allows modeling of poly and amorphous semiconductor devices such as TFTs in 3D. This model has all the functionality of the 2D TFT simulator.

e. Quantum3D

- This allows modeling of the effects of quantum confinement using the quantum moment approach. This model has all the functionality of the 2D Quantum model.

2. Orchid

- **Orchid** is a new product used in conjunction with **SPisces** to model the effects of total radiation dosage on MOS device reliability. Key to this capability is the ability to model the gate oxide as a semiconductor. The user may then define carrier pair generation in the oxide as a function of position, time, and electric fields, using a new C interpreter function. Another C interpreter function can be used to calculate the cumulative fixed charge density as a function of position and electron and hole current densities in the oxide. The deduced fixed oxide charge density directly effects the threshold degradation of the device.

3. Introduced SiC

- SiC is a new simulation capability for simulating anisotropic materials such as silicon carbide. The basis of this model is the incorporation of a new anisotropic mobility model that allows users to define separate mobilities along normal directions.
- Also added a new default value for thermal conductivity and electron saturation velocity of silicon carbide.

4. Modifications to Laser

Included additional functionality into Laser. The new features are:

- transient mode (time-dependent photon rate equation)
- far field patterns
- addtitional loss mechanisms including:
  - mirror loss
  - free carrier loss
  - absorption loss (including C interpreter function for bulk absorption)

5. New Large Lignal and Simulation Capabilities

- Added specifications of sinusoidal waveforms to the SOLVE statement.
- Implemented post processing Fourier analysis. This performs a Fast Fourier Transform (FFT) on data within a log file to convert time domain data into frequency domain data.

- Added solution compliance based on sinusoidal analysis. In this mode, transient simulation will stop if the amplitude of a sinusoidal waveform is less than the user-specified value.

6. Modifications to LUMINOUS

- Added capability to print the complex indices of refraction as they are calculated for the various materials during the ray trace in `Luminous`.
- Added functionality to simulate single-layer antireflective coatings in `Luminous`. This capability calculates the reflection coefficient for a single layer coating under conditions of normal incidence.

7. Modifications to MIXEDMODE

- Added the capability to specify multiple `.TRAN` statements in a single input deck.

Added the extraction of network parameters (`S, Z, Y, H, ABCD`), stability factor, unilateral power gain, and maximum unilateral transducer power gain in MIXEDMODE.

- Added the capability to use the PROBE in MIXEDMODE.

8. Modification to GIGA

- Enabled simulations of floating gate structures with heat flow in GIGA.

9. New Physical models

- Added Klaassen's concentration and temperature-dependent SRH lifetime and Auger rate models.
- Implemented new continuous trap level model in TFT.
- Added new carrier concentration dependent Auger coefficient model.
- Added the capability to calculate the band-to-band tunneling coeffiicients from principles.
- Added the capability to model thermionic field emission tunneling in Schottky contacts.
- Added two new parameters to the single-event upset model. The generated electron/holes pairs (or charge) will not be calculated after `tf` seconds if this parameter is specified in the input deck. The width of the beam can now be specified as a constant value, followed by an exponential or Gaussian function by using the `beam.radius` parameter. For radi greater than `beam.radius`, either the Gaussian or exponential functions is used (if they are specified).

10. New Numerical Techniques

- Two sparse matrix reordering techniques are now available: Minimum Degree (MD) and Reverse Cuthill McKee (RCM). They are used in the 2D linear solver. MD is the default. Choosing RCM instead of MD sometimes solves the problem of having a zero on the diagonal, which is indicated by the error message:

```
Internal error in linear solver:  Z-#.
```

  If this happens, the user should switch from MD to RCM or vice-versa.

- For Parallel ATLAS, two new mesh partitioning techniques are now available:

  – Natural Ordering (NO) and

  – Multi-level Graph Partitioning (MGP)

  These are used to assign sub-meshes to different processor. MGP is the default and should always outperform NO. On the other hand, NO requires less preprocessing time and might perform as well as MGP in the case of meshes generated by the ATLAS statement.

- Made it possible to use projection for initial guesses in conjunction with current boundaries.

11. Structures and Log File Improvements

- Added the capability to write X and G (rhs) errors in a standard structure file for subsequent visualization using TONYPLOT.
- Added the capability to save electron, hole relaxation times, and Peltier coefficients to

structure files.

- Added the capability to save electron, hole, and displacement currents to log files.

12. New Quantum models.

- Implemented a new self-consistent Schrodinger-Poisson solver. This model interactively solves Poisson's and Schrodinger's equations to calculate the potential and carrier concentration, including the effects of quantum confinement. These solutions also produce eigen energies and eigen functions.
- Added Hansch's model for quantum mechanical correction for carrier distributions in N channel MOSFETs inversion layers.
- Added van Dort's model for quantum mechanical correction for carrier distributions in N and P channel MOSFETs in inversion and accumulation layers.

13. Miscellaneous ATLAS **Improvements**

- Reconfigured small signal AC post processing to include the effects of parasitics in gain calculations.
- Changed functionality of C interpreter functions, `F.CONMUN` and `F.CONMUP`, to account for donor and acceptor concentrations separately.
- Added trap concentrations to concentration-dependent mobility models and band gap narrowing models.
- Added new C interpreter functions for energy-dependent relaxation times for electrons and holes.
- Added new C interpreter functions for energy-dependent Peltier coefficients for electrons and holes.
- Added a new C interpreter function to specify doping in 3D as a function of position.
- Added extra checking to the AC memory allocation. If there is insufficient memory for the factorization, more will be allocated. A warning message will also be printed.
- Enabled multiple contacts with the same name to have the same work function and other parameters.
- Modified the DOPING statement so that ATHENA master files can now be used in 2D. ATHENA doping is interpolated onto the ATLAS mesh.

Version 4.3.0.R

1. Introduced Quantum Transport Model

This model accounts for redistribution of carriers near abrupt material transitions due to the effects of quantum confinement. It is also useful when simulating heterojunction devices such as HBTs and HEMTs, as well as short channel thin gate oxide MOS devices.

2. Introduced Improvements to the Ferroelectric Model

This release includes modifications which allow better modeling of unsaturated loop simulation of ferroelectric materials. It also includes the capability to characterize the AC ferroelectric permitibity affected by the polarization as a fraction of the DC case.

3. Log File Improvements have been Made for Capturing Data

- PROBE has been enhanced to add the capability of probing minimums and maximums of the directional quantities; electric field, electron mobility, hole mobility and polarization.
- PROBE has been enhanced to allow the measurement of dielectric permitivity and the band-to-band tunneling rate.
- Added the capability to capture simulation time in the log file.

4. Structure File Improvements

The default method of averaging for capture of impact ionization rate, the electron and hole mobilities to the structure file has been changed. The new averaging techniques provide smoother plots in **TonyPlot**.

5. Improved Parameterization of Band-to-Band Tunneling Models.

This release allows user accessibility to the gamma parameter of the standard band-to-band tunneling model.

Version 4.0.0.R

1. Introduced a ferroelectric permitivity model

This model accounts for ferroelectric polarization and hysteresis. This release includes a new model to model ferroelectric materials. It is useful for simulating ferroelectric devices such as ferroelectric capacitors and FETs.

2. Introduced Klaassen's mobility model

This release includes a new model for low field electron and hole mobilities in silicon. This model accounts for lattice, impurity, and carrier scattering and screening as a function of temperature, doping and carrier concentrations. This model has been calibrated over a wide range of conditions and should be used as the default low field model for silicon simulations.

3. Introduced Concannon's gate and substrate current models

This release includes a new model for EPROM simulation. This model estimates the gate current in EPROM devices as a function of carrier temperature by integrating a non-maxwellian distribution function. The model also uses a impact ionization model that uses the same distribution function to ensure a direct relationship between gate and substrate currents.

4. Introduced Shirahata's mobility model

This release includes a new surface mobility model which accounts for carrier screening effects in the channel. This is done largely by using Klaassen's model for the low field mobility. The model also accounts for parallel and perpendicular fields in the channel.

5. Introduced a "Probe" capability

A new statement has been implemented which allows users to write local scalar quantities (e.g. potential, electron concentration, hole concentration, etc.) and vectors (e.g. field etc.) to the log file as a function of bias, time etc. The user may specify the probe by location, minimum or maximum, and type.

6. Modifications to the Watt mobility model

Several modifications were made to the Watt mobility model. First, the user may now apply the model to grid nodes below the interface down to a user specified depth. Also the user can use an exponential scaling on the effective field. Finally, the user can make the models only apply to minority carriers.

7. Improved vector averaging

Certain quantities used in ATLAS are not defined on grid points but between points (e.g. electric fields). To write these quantities into the structure files they must be averaged. In this implementation an improved averaging scheme was introduced which produces smoother results. The user also has control over various weighting schemes for the averaging.

8. Traps in MIXEDMODE

New parameters were introduced to the TRAP, INTTRAP, DEFECT, and INTERFACE statements to enable proper handling of these physical models in MIXEDMODE.

9. Modified run time outpu

The ATLAS run time output was modified to be more concise and informative.

10. Misalignment and delta CD in INTERCONNECT3D

INTERCONNECT3D now allows users to specify misalignment and variations in critical dimensions with interconnects specified by mask sets from MASKVIEWS.

11. Lumped elements and current boundary conditions in DEVICE3D

DEVICE3D now supports lumped element and current boundary conditions.

12. GaAs in DEVICE3D

DEVICE3D now supports GaAs material models.

13. Electric field lines.

Electric field lines can now be plotted in TONYPLOT. The specification of where to elines lines are calculated was improved for both plotting and for evaluation of ionization integrals.

14. Energy dependent impact ionization coefficients.

Two new C interpreter functions were introduced to enable user specification of impact ionization coefficients as a function of carrier temperature for Selberherr's model.

15. Removed obsolete syntax

The `FIT`, `COLOR`, `CONTOUR`, `LABEL`, `LOOP`, `L.END`, `PLOT.1D`, `PLOT.2D` and `PLOT.3d` statements are no longer supported by ATLAS. Equivalent functionalities are available through DECKBUILD and TONYPLOT.

16. DEVICE3D improvements for SEU simulation

Introduced a new C interpreter function, `F3.RADIATE`. This function allows users to specify arbitrary generation rate versus spatial position and time. The `SINGLEEVENTUPSET` statement was changed to allow users to control whether the generation rate is scaled by the ratio of the numerical and analytic integrals of the analytic generation function. This permits conservation of total generation rate in a device. By default the rescaling is now turned off.

17. MIXEDMODE **Improvements**

The .DC statement now allows linear and logarithmic stepping of sources and nested sweeps.

Version 3.0.0.R

1. First release of "six equation solver" version of ATLAS**.**

In this release an advanced Non-Isothermal Energy Balance Model was implemented. The model allows self-consistent solution of a set of up to 6 partial differential equations for electrostatic potential, electron and hole concentrations, electron and hole carrier temperatures, and lattice temperature. All the important combinations of equations are supported.

2. DEVICE3D **introduced.**

DEVICE3D is introduced as the first truly 3D simulator in the ATLAS framework. It is a virtual analogy to S-PISCES. With it three dimensional steady-state silicon device simulation can be performed to calculate electrical behavior of bipolar and MOS transistors, as well as EPROM devices.

- Syntactically and behaviorally like ATLAS S-PISCES.
- Poisson, one carrier or two carrier analysis can be performed.
- Uses Gummel, Newton or combined Gummel/Newton algorithms.
- Supports semi-implicit scheme for transient simulation. Significantly faster than TR-BDF.

- Supports the essential set of physical models for silicon modeling including: SRH recombination,

Auger recombination and radiative recombination; concentration dependent mobility models (Conmob,Arora, Analytical); longitudinal and transverse electric field dependent mobility models (Caughey-Thomas, CVT, Yamaguchi); Band Gap Narrowing (BGN), Boltzman statistics, Fermi-Dirac statistics, and incomplete ionization; Fowler-Nordheim current calculation, Lucky electron model for hot carrier injection and transient programming and erasing of EPROM devices.

- The three dimensional simulated structure can be specified using DEVEDIT3D.
- Uses TONYPLOT3D for postprocessing.
- Single Event Upset simulation is feasible with DEVICE3D. The capability of single event upset/ photogeneration transient simulation has been introduced. It allows specification of multiple tracks and radial, length and time dependence specification of generated charge along tracks.

3. THERMAL3D **introduced.**

THERMAL3D is introduced as a new capability for three dimensional simulation using ATLAS. The Poisson equation for temperature is solved in three dimensional regions of specified heat conductivity with defined temperatures on thermal contacts (which are ideal heat conductors). The user may specify the thermal properties of regions heat sources, and heat sinks.

- Four models of temperature dependent thermal conductivity.
- User may specify independent temperatures on all thermal sinks (thermal electrodes).
- User may specify multiple independent heat sources.
- Completely compatible with ATLAS.
- The simulated structure is specified using DEVEDIT3D (including thermal electrodes).
- Temperature distributions can be examined using TONYPLOT3D.

4. INTERCONNECT3D **introduced**

INTERCONNECT3D has been introduced as another 3D capability in the ATLAS framework. This simulator is used to extract parasitics from arbitrary 3D interconnect structures.

- Can extract parasitic capacitance and conductance for arbitrary 3D interconnect structures.
- Provides an interface to MASKVIEWS for simple mask layer definition of interconnect structure.
- Completely compatible with ATLAS.

5. Curve tracer introduced

Allows automatic tracing of complex IV behavior (including functions that are multivalued) without user intervention. Automatically varies an external resistance on a given contact to ensure optimal steps are chosen.

6. Enhancements for Energy Balance Model

Aside from the introduction of the “Six Equation Solver”, several other improvements were made with respect to the Energy Balance Model (EBM).

- Fermi statistics can be used with the EBM.
- The “GaAs” like mobility model can be used with the EBM.
- The energy relaxation length can be specified for impact ionization in the EBM.
- Carrier temperature projection for initial guesses is supported.

7. New solution methods

8. GIGA **improvements**

Aside from the introduction of the “Six Equation Solver”, several other enhancements were made to GIGA.

- GIGA now allows self consistent 5 and 6 equation solutions of lattice temperature with electron and hole temperatures.
- GIGA now supports static projection for initial guesses for lattice temperature.

9. LUMINOUS **improvements**

In this release many improvements were made to LUMINOUS.

- LUMINOUS now supports extraction of spatial response to optical sources.
- A MIN.POWER parameter of the BEAM statement can now be used to limit the number of rays traced in LUMINOUS.
- LUMINOUS now recognizes metals (electrodes) as bulk regions.
- Optical properties of bulk metal (electrodes) region can now be defined.
- The C Interpreter function RADIATE for user definable photogeneration was enhanced to include time dependence.

10. TFT improvements

Several enhancements were made in this release for TFT.

- Parameters were introduced to allow control of the number of mid-gap states used for TFT simulations.
- TFT now supports output of donor and acceptor trap state densities and energies to user specifiable files for examination by TONYPLOT.
- TFT now provides two new Interpreter functions for specifying the donor and acceptor densities as a function of energy.

11. Miscellaneous new features

A host of new features were introduced into ATLAS in general.

ATLAS now properly handles touching electrodes and electrodes with the same name.

Incorporated the trap assisted tunneling model given in Hurkx, G.A.M., Klassen, D.B.M. and Knuvers, M.P.G., "A New Recombination Model for Device Simulation Including Tunneling", IEEE Transactions on Electron Devices, V. 39, No. 2. Feb., 1992, pp. 331-338. This model accounts for tunneling via trap transitions using an analytic approximation that modifies the lifetimes in the standard Schockley-Read-Hall recombination model.

Modified the models for hot electrons, and holes and Fowler Nordheim tunneling currents. The new models account for the flow of the electrons/holes in the insulating layers to arbitrary electrodes based on one of two primitive models. The first, is the default model and trys to force the currents to follow the maximum potential gradients to find the proper electrode. In the second model insulator, currents go the the nearest electrode. With these modifications the electrode current run time outputs are modified to better reflect the new models.

Enabled correct small signal simulation when simulating with bulk and interface traps.

Distinction made between active and chemical dopant densities.

Small signal analysis with interface traps now works properly.

Moved the C Interpreter function F.COMPOSIT from the MATERIAL statement to the DOPING statement. In previous releases this function did not work as advertised, since material composition needed to be defined at structure creation. In order to make the function work properly it had to be moved to the DOPING statement.

In addition to CLIMIT parameter which existed earlier, new parameters CLIM.DD and CLIM.EB were added in METHOD statement. CLIM.DD is exact analog of CLIMIT parameter, but in comparison with CLIMIT it specifies dimensional value of minimal concentration which can be resolve using ATLAS. By default CLIM.DD is not specified and ATLAS uses default value of CLIMIT, as in previous versions. Parameter CLIM.EB can be treated as regularization parameter for the case of very small electron (hole) concentration for energy balance model. It specifies the mimimal value of concentration for which relaxation term in energy balance equation will be still properly resolved, otherwise

temperature for the points where concentration is much less than CLIM.EB, will tend to lattice temperature.

12. New syntax

With this release, the ATLAS syntax was augmented to simplify and somewhat rationalize some of the more confusing problems.

- The SYMBOLIC statement is no longer needed.
- The numerical method can now be specified on the MODEL statement. The user may choose from Gummel, Block and Newton methods. More than one method can be chosen; and ATLAS provides automatic switching between the methods.
- ATLAS now supports a new syntax for specifying which equations are simulated. The syntax is now supported on either the MODEL or METHOD statement and is the preferred syntax for future compatibility. The new syntax provides a more rational, intuitive way of choosing which equations are solved.
- ATLAS still supports syntax from previous versions.

13. Improved run-time output

Many improvements have been made to the run-time outputs of ATLAS.

- Added multiple levels of verbosity.
- Improved printing of mesh statistics, region and electrode summaries and models.
- Printing of error messages was made more consistent and informative in many places in ATLAS.
- Removed spurious printout from S parameter extraction with the UTMOST statement.

14. New platforms

ATLAS has now been ported to DEC Alpha and SGI workstations.

15. Miscellaneous bug fixes

In this release many bugs were removed that existed in the previous version.

- Fixed bug that caused setting the value of the MODEL statement parameter CCS.EA to also set the value of the MATERIAL statement parameter COPTR.
- Fixed bug causing improper association of nodes with regions on BLAZE structures read in from DevEdit. In some instances nodes j would be assigned a region at the interface then the composition fraction (e.g.) from the opposing region, thus causing distortions in the band diagram at the interface.
- Fixed bug in static projection. Bug caused by two consecutive bias steps of opposite polarity.
- Fixed bug in TFT. Caused the midgap Gaussian donor defects to have the same distribution as the midgap acceptor defects.
- Fixed bug in UTMOST interface. Bug caused ATLAS not to recognize routines 2, 3, 5, 12, 13, 14, and 26.
- The calculations for the electron and hole mobilities in the InGaAsP system were found to be in error both in the manual as well as the code. Referring to the ATLAS Users manual, Edition 2, in the equations B-36 and B-37, the “y”s should be replaced by (1-y). This fix was introduced into the code for this release.
- Fixed a bug in the calculation of electric fields for output to structure files for examination in TONYPLOT. This bug caused the estimates of the X and Y components of the electric field to be calculated based on the dielectric constant of silicon for all semiconductors. Although, the calculation properly handled insulators it used the dielectric constant of silicon for all semiconductors regardless of composition. This has been fixed in this version.

  Fixed bug causing extra space in UTMOST format output log files.

16. New examples

In this release the standard examples were completely reworked and improved. The ATLAS examples are now divided into 21 application specific areas. There are now 117 ATLAS examples in all which illustrate virtually all the functionality of ATLAS.

17. Improved quality control

In this version, new quality control tools and procedures for automated testing were implemented and used that greatly simplify and expanded ATLAS testing. This version of ATLAS represents the most thoroughly tested version to date. Continued application of the improved quality control procedures will ensure that future versions will continue to be of the highest quality.

Version 2.0.0.R

1. **Improved Standard Structure Output** – The standard structure output format has been improved to account for multiple values at region interfaces (e.g., contours projecting into regions where they did not apply).
2. **Reliable Metal Regions** – Fixed bug that caused metal regions on standard structure format structures that are read into ATLAS to be lost following modification in ATLAS by either addition of electrodes or regridding.
3. **MEASURE Statement** – Because of conflicts with the DECKBUILD extract compatibility syntax, the ATLAS EXTRACT statement has been changed to MEASURE.
4. **Error Messages** – Fixed bug that caused infinite loop of error messages during write of structures containing isolated triangles.
5. **Larger Allocation** – Fixed bug for "large" version (altas-E) for dynamic allocation to allow up to 6,000 nodes.
6. **Ramping** – Fixed bug for case where frequency is ramped in AC analysis during a static ramp. The frequency is now properly reset to its base value after each static solution.
7. **Maximum Time Step** – Added DT.MAX parameter to the METHOD statement to limit the maximum time step size during transient solutions.
8. **Electron Current Vectors** – Fixed bug that caused electron current vectors to point in the wrong direction in TONYPLOT.
9. **Ionization Printout** – Removed spurious printout during ionization integral calculation.
10. **Incomplete Ionization** – Fixed bug in incomplete ionization. Bug caused lack of convergence.
11. **Interpreter** – Fixed bug in F.DOPING Interpreter function for specifying doping.
12. **Material** – Fixed bug to allow Interpreter functions in material statement to be defined for a given material.
13. **Region Interface** – Fixed bug to resolve ambiguities in definition of region membership of nodes at interface between two regions.
14. **Holes** – Fixed bug that caused holes in regions or embedded regions to be improperly displayed in **TonyPlot**.
15. **Trap Recombination** – Fixed bug in recombination calculation for regionally specified traps.
16. **Transient Projection Algorithm –** Installed transient projection algorithm.
17. **Improved Interpreter Function for Complex Index of Refraction –** The C Interpreter function for complex index of refraction was changed to account for material composition and temperature.

18. **New Built-in Index Data –** Added built-in complex index of refraction data for: AlAs, GaAs, InSb, InP, Polysilicon, and Si02.
19. **Light Intensity Contours** – Added interface to TONYPLOT for optical intensity. To enable, set the OPT.INTENS parameter of the OUTPUT statement.
20. **New C Interpreter Function –** Added C Interpreter function for general generation rate as a function of position.
21. **Cylindrical Coordinates – LUMINOUS** now supports cylindrical coordinates. Specify CYL on the MESH statement.
22. **Energy Balance Modeling – LUMINOUS** is now supported in energy balance simulations.
23. **Loading and Saving Optical Sources –** LOAD and SAVE statements act to save and restore optical sources.

BLAZE Version 2.0.0.R

24. **Improved DevEdit Composition:** Fixed bug that caused DEVEDIT composition fraction information to be lost when the DEVEDIT structure is loaded in MESH statement followed by any ELECTRODE statement.
25. **Composition Fraction Modification:** Fixed bug to allow composition fraction modification on loaded structures.
26. **Thermionic Emission:** Added capability to specify thermionic emission for heterojunctions in loaded structures.

GIGA Version 2.0.0.R

27. **Quasi-Fermi Level:** Fixed bug that caused improper display of the hole quasi-fermi level in **GIGA** generated structure files.
28. **Outer Block Iteration:** Fixed bug in GIGA that caused premature exit from the outer block iteration when the maximum temperature update exceeded the starting temperature during the initial solution.
29. **Material Thermal Parameters:** The ability to specify the GIGA thermal parameters dependently for different materials/regions was added in this version.
30. **Improved Thermal Dependencies:** The temperature dependencies of many models were improved.

LASER Version 2.0.0.R (Initial Release of LASER Under The ATLAS Framework)

31. This version supports simulation of various types of Fabry-Perot semiconductor diode lasers.

BLAZE Version 2.0.0.R

32. Now supports running MIXEDMODE decks under DECKBUILD.
33. Implemented small-signal AC simulation in MIXEDMODE.
34. **Energy Balance for Heterojunctions:** Modifications to to the ATLAS energy balance formulas were introduced to account for positionally dependent band structures.
35. **Improved Material Defaults:** Some material model parameter defaults were changed to reflect better models in literature. In particular, these improvements affect InGaAsP and related compounds, AlGaAs, and SiGe.
36. **Improved DevEdit Interface:** Donor and acceptor impurities were added to the DEVEDIT interface for doping specification in DEVEDIT.

MIXEDMODE Version 2.0.0.R

37. User-definition of how often solutions are saved during transient simulation

38. The cylindrical coordinate system may now be used with MIXEDMODE.

39. Schottky contacts are now supported in MIXEDMODE.

40. Tabulated time-dependent voltage and current sources may be specified.

41. Previously calculated device solutions can be loaded into MIXEDMODE.

42. Circuits may include lossless transmission lines.

43. User-defined two-terminal elements may be specified.

TFT Version 2.0.0.R

44. User Specifiable Thermal Conductivities of Metals: GIGA now recognizes metals (electrodes) as different materials from semiconductors and insulators. Users can now specify metal thermal conductivity.

45. Regional DOS Specification: Added `REGION` parameter to the `DEFECT` statement to allow DOS to be defined differently in different regions.

Bibliography

1. C.R. Crowell, S.M. Sze, “Temperature Dependence of Avalanche Multiplication in Semiconductors”, Applied Physics Letters 9, pp. 242-244, 1966.
2. W.N. Grant, “Electron and Hole Ionization Rates in Epitaxial Silicon at High Electric Fields”, Solid-State Electronics, 16, pp. 1189-1203, 1973.
3. S. Selberherr, “Analysis and Simulation of Semiconductor Devices”, Springer-Verlag, Wien-New York. 1984.
4. M.R. Pinto, Conor S. Rafferty, and Robert W. Dutton, “PISCES2 - Poisson and Continuity Equation Solver”, Stanford Electronics Laboratory Technical Report, Stanford University, September 1984.
5. J.T. Watt, Ph.D. Thesis, Stanford University, 1989.
6. S.A. Schwarz and S.E. Russe, “Semi-Empirical Equations for Electron Velocity in Silicon: Part II — MOS Inversion Layer”, IEEE Trans. on Electron Devices. ED-30, N12, pp. 1634-1639, 1983.
7. H. Shin, A.F. Tasch, C.M. Maziar, and S.K. Banerjee. “A New Approach to Verify and Derive a Transverse-field-dependent Mobility Model for Electrons in MOS Inversion Layers”, IEEE Trans. Electron Devices, ED-36 No. 6, pp. 1117-1123, 1989.
8. J.T. Watt, and J.D. Plummer, “Universal Mobility-Field Curves for Electrons and Holes in MOS Inversion Layers”, presented at 1987 Symposium on VLSI Technology, Karuizawa, Japan.
9. H. Shin, G.M. Yeric, A.F. Tasch, and C.M. Maziar. “Physically-based Models for Effective Mobility and Local-field Mobility of Electrons in MOS Inversion Layers”, to be published.
10. S.P. Gaur and D.H. Navon, “Two-Dimensional Carrier Flow in a Transistor Structure Under Nonisothermal Conditions”, IEEE Trans. on Electron Devices. ED-23, N1, pp. 50-57, 1976.
11. D.J. Roulston, N.D. Arora, and S.G. Chamberlain, “Modeling and Measurement of Minority-Carrier Lifetime versus Doping in Diffused Layers of n ±p Silicon Diodes”, IEEE Trans. on Electron Devices, ED-29, pp. 284-291. Feb., 1982.
12. A.S. Grove, “Physics and Technology of Semiconductor Devices”, Wiley, 1967.
13. S.M. Sze, “Physics of Semiconductor Devices”, Wiley, 1981.
14. J.W. Slotboom, “The PN Product in Silicon”, Solid State Electronics 20, pp. 279-283. 1977.
15. Z. Yu, Ph.D. dissertation, Stanford Electronic Laboratories, Stanford University.
16. Z. Yu and R.W. Dutton, “SEDAN III - A Generalized Electronic Material Device Analysis Program”, Stanford Electronics Laboratory Technical Report, Stanford University, July 1985.
17. W.B. Joyce and R. W. Dixon, “Analytic Approximation for the Fermi Energy of an ideal Fermi Gas”, Appl. Phys Lett. 31, pp. 354-356, 1977.
18. R.C. Jaeger and F. H. Gaensslen, “Simulation of Impurity Freezeout through Numerical Solution of Poisson’s Equations and Application to MOS Device Behavior”, IEEE Trans. on Electron Devices, ED-27, pp. 914-920, May 1980.
19. R. Van Overstraeten and H. Deman, “Measurement of the Ionization Rates in Diffused Silicon p-n Junctions”, Solid-State Electronics, 13, pp. 583-608. 1970.
20. W.T. White, et.al., IEEE Trans. on Electron Devices. ED-37, p. 2532, 1990.
21. O.S. Heavens, “Optical Properties of Thin Solid Films”, Dover, New York,
22. D.M. Caughey and R. E. Thomas, “Carrier Mobilities in Silicon Empirically Related to Doping

and Field", Proc. IEEE 55, pp. 2192-2193. 1967.

23. S. Selberherr, "Process and Device Modeling for VLSI", Microelectron. Reliab. 24 No. 2, pp. 225-257, 1984.
24. J.J. Barnes, R.J. Lomax, and G.I. Haddad, "Finite-element Simulation of GaAs MESFET's with Lateral Doping Profiles and Sub-micron Gates", IEEE Trans. on Electron Devices, ED-23, pp. 1042-1048, September, 1976.
25. M.J. Littlejohn, J.R. Hauser, and T.H. Glisson, "Velocity-field Characteristics of GaAs with $\Gamma_6^e - L_6^e - X_6^e$ Conduction Band Ordering", J. Applied Physics 48, no. 11, pp. 4587-4590, November 1977.
26. D.H. Rose and R.E. Bank, "Global Approximate Newton Methods", Numerische Mathematik 37, pp. 279-295, 1981.
27. T. Wimp, "Sequence Transformation and their Application", Academic Press, 1981.
28. D.G. Anderson, "Iterative Procedure for Nonlinear Integral Equations", ACM Journal, Vol. 12, No. 4, pp. 547-560.
29. H. Iwai, M.R. Pinto, C.S. Rafferty, J.E. Oristian, and R.W. Dutton, "Velocity Saturation Effect on Short Channel MOS Transitor Capacitance", IEEE Electron Device Letters, EDL-6, pp. 120-122, March 1985.
30. S. Tam, P-K Ko, and C. Hu, "Lucky-electron Model of Channel Hot-electron Injection in MOSFET's". IEEE Trans. on Electron Devices, Vol. ED-31, N9, September 1984.
31. C.R. Crowell and S. M. Sze, "Current Transport in Metal-Semiconductor Barriers", Solid State Electronics, 9, pp. 1035-1048, 1966.
32. E. Sangiorgi, C.S. Rafferty, M.R. Pinto, and R.W. Dutton, "Non-planar Schottky Device Analysis and Applications", Proc. International Conference on Simulation of Semiconductor Devices and Processes, Swansea, U.K., July 1984.
33. C.S. Rafferty, M.R. Pinto, and R.W. Dutton, "Iterative Methods in Semiconductor Device Simulation", IEEE Trans. on Electron Devices, ED-32, pp. 2018-2027, October 1985.
34. R.S. Varga, "Matrix Iterative Analysis", Prentice-Hall, Englewood Cliffs, NJ, 1962.
35. D.L. Scharfetter and H.K. Gummel, "Large-Signal Analysis of a Silicon Read Diode Oscillator", IEEE Trans. on Electron Devices, ED-16, pp. 64-77, January 1969.
36. C.H. Price, "Two Dimensional Numerical Simulation of Semiconductor Devices", Ph.D. Dissertation, Stanford University, May 1982.
37. B.A. Carre, Computer Journal 4, 1961, p. 73.
38. E.M. Buturla and P.E. Cotrell, "Simulation of Semiconductor Transport Using Coupled and Decoupled Solution Techniques", Solid State Electronics 23, No. 4, pp. 331-334, 1980.
39. S.C. Eisenstat, M.C. Gursky, M.H. Schultz, A.H. Sherman, Computer Science Department, Yale Univ., Tech. Rep. 112, 1977.
40. R.E. Bank and A.H. Sherman, "A Refinement Algorithm and Dynamic Data Structure for Finite Element Meshes", University of Texas at Austin Technical Report CS TR-159/CNA Tr-166, October 1980.
41. B.S. Baker, E.M. Grosse, and C.S. Rafferty, "Non-obtuse Triangulation of General Polygons", to be published.
42. C.W. Gear, "Numerical Initial Value Problems in Ordinary Differential Equations", Prentice-Hall, 1971.
43. A. De Mari, "An Accurate Numerical One-dimensional Solution of the P-N Junction Under Arbitrary Transient Conditions", Solid-State Electronics 11, pp. 1021-1053, 1968.

44. J. Lambert, “Computational Methods in Ordinary Differential Equations”, Wiley, 1973.

45. R. Bank, W.M. Coughran, W. Fichtner, E.H. Grosse, D.J. Rose, and R.K. Smith, “Transient Simulation of Silicon Devices and Circuits”, IEEE Trans. on Electron Devices, ED-32, pp. 1992-2007, October 1985.

46. S.E. Laux, “Techniques for Small-Signal Analysis of Semiconductor Devices”, IEEE Trans. on Electron Devices, ED-32, pp. 2028-2037, October 1985.

47. B. Polsky and J. Rimshans, “Half-Implicit Scheme for Numerical Simulation of Transient Processes in Semiconductor Devices”, Solid-State Electronics, Vol. 29, pp. 321-328, 1986.

48. R. Stratton, “Phys. Rev.”, 126 (6) p. 2002, 1962.

49. K. Blotekjaer, IEEE Trans., ED 17, p. 38, 1970.

50. B. Meinerzhagen and W.L. Engl, “The Influence of Thermal Equilibrium Approximation on the Accuracy of Classical Two-Dimensional Numerical Modeling of Silicon Submicrometer MOS Transistors”, IEEE Trans., ED-35, No. 5, pp. 689-697, 1988.

51. K. Katayama and T. Toyabe, “A New Hot Carrier Simulation Method Based on Full 3D Hydrodynamic Equations”, IEDM-89, Technical Digest, p. 135, 1989.

52. G. Liu and S. Fonash, “Low Thermal Budget Poly-Si Transistors on Glass”, JJAP, Vol. 30, p. 269, 1991.

53. K. Fujii, Y. Tanaka, K. Honda, H. Tsutsu, H. Koseki, and S. Hotta, “Process Techniques of 15 inch Full Color and High Resolution a-Si TFT LCD”, 5th Int. MicroProcess Conf., Kawasaki, Japan, 1992.

54. F. Hayashi and M. Kitakata, “A High Performance Polysilicon TFT Using RTA and Plasma Hydrogenation Applicable to Highly Stable SRAM’s of 16 Mbit and Beyond”, Digest of VLSI Technology Symposium, p. 36, 1992.

55. N. Lifshitz, S. Fang, and A. Kornblit, “Novel Thin Film Transistor for SRAM Applications”, Proceedings International Semiconductor Device Research Symposium, Charlottesville, VA, p. 529, 1991.

56. R. Ditizio, S. Fonash, and B. C. Hshieh, “Examination of The Optimization of TFT Passivation with Hydrogen ECR Plasma”, J. Vacuum Science and Technology, Vol. 10, p. 59, 1992.

57. M. Hack, H. Tuan, J. Shaw, M. Shur, and P. Yap, “Physics of Novel Amorphous Silicon High-Voltage Transistor”, Mat. Res. Soc. Symp. Proc. Vol. 95, p. 457, 1987.

58. T. Shaw and M. Hack, “An Analytical Model for Calculating Trapped Charge in Amorphous Silicon”, Journal of Applied Physics, 6419, p. 4562, 1988.

59. J. Smith, Ph.D. Thesis, Pennsylvania State University, 1992.

60. G.K. Wachutka, “Rigorous Thermodynamic Treatment of Heat Generation in Semiconductor Device Modeling”, IEEE Trans., Computer-Aided Design, Vol. 9, No. 11, pp. 1141-1149, 1990.

61. R. Stratton, “Semiconductor Current-Flow Equations (Diffusion and Degeneracy)”, IEEE Trans., ED, Vol. ED-19, No. 12, pp. 1288-1292, 1972.

62. A. Nakagawa and H. Ohashi, “A Study on GTO Turn-off Failure Mechanism — A Time- and Temperature-Dependent 1-D Model Analysis”, IEEE Trans., ED, Vol. ED-31, No. 3, pp. 273-279, 1984.

63. P. Mars, “Temperature Dependence of Avalanche Breakdown Voltage in p-n Junctions”, International Journal of Electronics, Vol. 32, No. 1, pp. 23-27.

64. Handbook of Chemistry and Physics, Editor R.C. Weast.

65. M. Klausmeier-Brown, M. Lundstrom, M. Melloch, “The Effects of Heavy Impurity Doping on AlGaAs/GaAs Bipolar Transistors”, IEEE Trans., ED-36, No. 10, pp. 2146-2155, 1989.

66. J. Dorkel and Ph. Leturcq, “Carrier Mobilities in Silicon Semi-Empirically Related to

Temperature Doping and Injection Level", Solid-State Electronics, 24, pp. 821-825, 1981.

67. G.A.M. Hurkx, H.C. de Graaf, W.J. Klosterman, et. al., "A Novel Compact Model Description of Reverse Biase Diode Characteristics including Tunneling", ESSDERC, pp. 49-52, 1990.
68. D.B.M. Klaassen, "Physical Modeling fro Bipolar Device Simulation", In: Simulation of Semiconductor Devices and Processes, Edited by W. Fichtner and D. Aemmer, Harting-Gorre, 1991, Vol. 4, pp. 23-43.
69. S. Keeney, F. Piccini, M. Morelli, A. Mathewson, et. al., "Complete Transient Simulation of Flash EEPROM Devices", IEDM Technical Digest, pp. 201-204, 1990.
70. J. Sutherland and F. Hauser, "A Computer Analysis of Heterojunction and Graded Composition Solar Cells", IEEE Trans., ED-24, pp. 363-373, 1977.
71. M. Lundstrom and R. Shuelke, "Numerical Analysis of Heterostructure Semiconductor Devices", IEEE Trans., ED-30, pp. 1151-1159, 1983.
72. C.M. Wu and E.S. Yang, "Carrier Transport Across Heterojunction Interfaces", Solid-State Electronics, 22, pp. 241-248, 1979.
73. R.L. Anderson, "Experiments on Ge-GaAs Heterojunctions", Solid State Electronics, Vol. 5, pp. 341-351, 1962.
74. Kernighan and Ritchie,"The C Programming Language", Prentice-Hall, 1978.
75. Adachi, S., Physical Properties of III-V Semiconductor Compounds InP, InAs, GaAs, GaP, InGaAs, and InGaAsP, John Wiley and Sons, New York, 1992.
76. Green, Martin A., Intrinsic Concentration, Effective Densities of States, and Effective Mass in Silicon, J. Appl. Phys., V. 67, No. 6, March 6, 1990.
77. Law, M.E., et. al., Self-Consistent Model of Minority-Carrier Lifetime, Diffusion Length, and Mobility, IEEE Electron Device Letters, Vol. 12, Nol 8, 1991.
78. Muller, R.S., and Kamins, T.I., Device Electronics for Integrated Circuits, 2nd Ed., John Wiley and Sons, New York, 1986.
79. Pejcinovic, B., et. al., Numerical Simulation of Semiconductor Devices, Springer-Verlag Wien, New York, 1984.
80. Sze, S. M., Physics of Semiconductor Devices, 2nd Ed., John Wiley and Sons, New York, 1981.
81. Yoshida, J., et. al., Emitter-Base Bandgap Grading Effects on GaAlAs/GaAs Heterojunction Bipolar Transistor Characteristics, IEEE Trans. on Elect. Devices, Vol. ED-32, No. 9, Sept. 1985.
82. Nomenclature of Inorganic Chemistry, Journal of the American Chemistry Society, No. 82, p. 5525, 1960.
83. K. Mayaram, "CODECS: A Mixed-Level Circuit and Device Simulator", Memo No. UCB/ERL M88/71, University of California, Berkeley, CA.
84. K. Mayaram and D.O. Pederson, "Coupling Algorithms for Mixed-Level Circuit and Device Simulation", IEEE Transactions on Computer- Aided Design, Vol. 11, No. 8, August 1992, pp. 1003-1012.
85. Semiconductor device modeling with SPICE, Ed. by P. Antogneti and G.Massobrio, McGraw-Hill Book Company, 1987.
86. UTMOST III Modeling Manual, SILVACO International, 1993.
87. Rhoderick, E.H., and Williams, R.H., "Metal-Semiconductor Contacts", Second Ed., Oxford Science Publications, 1988.
88. W. Hansch, et al., "A New Self-Consistent Modeling Approach to Investigating MOSFET Degradation", IEEE Trans., ED-11, p. 362, 1990.
89. S. Odanaka, A. Hiroki, "A Numerical Simulation of Hot-Carrier Induced Device Degradation".

90. P. Roblin, A. Samman, S. Bibyk, “Simulation of Hot-Electron Trapping and Aging of nMOSFET’s”, IEEE Trans., ED-35, p. 2229, 1988.

91. D.P. Wilt and A. Yariv, “A Self-Consistent Static Model of the Double-Heterostructure Laser”, IEEE Journal of Quantum Electronics, Vol. QE-17, No. 9, 1981, pp. 1941-1949.

92. K.B. Kahen, “Two-Dimensional Simulation of Laser Diodes in Steady State”, IEEE Journal of Quantum Electronics, Vol. 24, No. 4, April 1988.

93. T. Ohtoshi, K. Yamaguchi, C. Nagaoka, T. Uda, Y. Murayama and N. Chinone, “A Two-Dimensional Device Simulator of Semiconductor Lasers”, Solid-State Electronics, Vol. 30, No. 6, pp. 627-638, 1987.

94. G. Hugh Song, K. Hell, T. Kerkhoven and U. Ravaioli, “Two-Dimensional Simulator for Semiconductor Lasers”, Proc. of the Int. IEEE Electron Device Metting, Washington, 1989, p. 143.

95. A. Yariv, Optical Electronics, CBS College Publishing, 1985.

96. S. Seki, T. Yamanaka and K. Yokoyama, “Two-Dimensional Analysis of Current Blocking Mechanism in InP Buried Heterostructure Lasers”, J. Appl. Phys. 71(7), April 1992, pp. 3572-3578.

97. G. Baccarani, M. Rudan, R. Guerrieri, and P. Ciampolini, “Physical Models for Numerical Device Simulation”, European School of Device Modeling, University of Bologna, 1991.

98. J.W. Slotboom and H.C. De Graaf, “Measurements of Bandgap Narrowing in Silicon Bipolar Transistors”, Solid State Electronics, Vol. 19, pp. 857-862, 1976.

99. J.G. Fossum and D.S. Lee, “A Physical Model for the Dependence of Carrier Lifetime on Doping Density in Nondegenerate Silicon”, Solid State Electronics, Vol. 25, pp. 741-747, 1982.

100. J. Dziewior and W. Schmid, “Auger Coefficient for Highly Doped and Highly Excited Silicon”, Appl. Phys. Lett., Vol. 31, pp. 346-348, 1977.

101. W.L. Engl and H.K. Dirks, “Models of Physical Parameters in an Introduction to the Numerical Analysis of Semiconductor Devices and Integrated Circuits”, J.J.H. Miller Ed., Dublin: Boole Press, 1981.

102. K.K. Thornber, “Relation of Drift Velocity to Low Field Mobility and High Field Saturation Velocity”, J. Appl. Phys., Vol. 51, pp. 2127- 2136, 1980.

103. K. Yamaguchi, “A Mobility Model for Carriers in the MOS Inversion Layer”, IEEE Trans. Elect. Devices, Vol. ED-30, pp. 658-663, 1983.

104. G.A.M. Hurkx, H.C. de Graaf, W.J. Klosterman, et. al., “A Novel Compact Model Description of Reverse Biased Diode Characteristics including Tunneling, ESSDERC, pp. 49-52, 1990.

105. D.B.M. Klaasen, “Physical Modeling for Bipolar Device Simulation — Simulation of Semiconductor Devices and Processes”, Edited by W. Fichter and D. Aemmer, Zurich, Harting-Gorre, 1991, Vol. 4, pp. 23-43.

106. G.A.M. Hurkx, D.B.M. Klaasen, M.P.G. Knuvers, and F.G. O’Hara, “A New Recombination Model Describing Heavy-Doping Effects and Low Temperature Behaviour”, IEDM Technical Digest, pp. 307-310, 1989.

107. A.I. Adamsone and B.S. Polsky, “3D Numerical Simulation of Transient Processes in Semiconductor Devices”, COMPEL — The International Journal for Computation and Mathematics in Electrical and Electronic Engineering, Vol. 10, No. 3, pp. 129-139.

108. Lombardi et al, “A Physically Based Mobility Model for Numerical Simulation of Non-Planar Devices”, IEEE Trans. on CAD, Nov. 1988, p. 1164.

109. B. Meinerzhagen, K. Bach, I. Bozk, and W.L. Eugl, “A New Highly Efficient Nonlinear

Relaxation Scheme for Hydrodynamic MOS Simulations", Proe. NUPAD IV, p. 91, 1992.

110. R.J. G. Gossens, S. Beebe, Z. Yu and R.W. Dutton, "An Automatic Biasing Scheme for Tracing Arbitrarily Shaped I-V curves", IEEE Trans., Computer-Aided Design, vol. 13, No. 3, pp. 310-317.

111. Shirahata M., Kusano H., Kotani N., Kusanoki S. and Akasaka Y., "A Mobility Model Including the Screening Effect in MOS Inversion Layer", IEEE Trans. Computer-Aided Design, V. 11, No. 9, pp. 1114-1119, Sept. 1992.

112. Concannon, A., Piccinini, F., Mathewson, A., and Lombardi, C., "The Numerical Simulation of Substrate and Gate Currents in MOS and EPROMs", (to be published).

113. Miller, et. al., "Device modeling of ferroelectric capacitors", J. Appl. Phys. 68 (12) 15 Dec. 1990.

114. Klaassen, D.B.M., "A Unified Mobility Model for Device Simulation - I. Model Equations and Concentration Dependence", Sol. St. Elec., V. 35, No. 7, pp. 953-959, 1992.

115. Klaassen, D.B.M., "A Unified Mobility Model for Device Simulation - II. Temperature Dependence of Carrier Mobility and Lifetime", Sol. St.Elec., V. 35, No. 7, pp. 961-967, 1992.

116. Klaassen, D. B. M., Slotboom, J. W. and De Graaff, H. C., "Unified Apparent Bandgap Narrowing in n- and p- type Silicon", Sol. St. Elec.,V. 35, No. 2, pp. 125-129, 1992.

117. Wang, "Fundamentals of Semiconductor Theory and Device Physics", Prentice Hall Inc. p. 521.

118. Yu, Z., Chen, D., So, L., and Dutton, R. W. "Pisces-2ET, Two Dimensional Device Simulation for Silicon and Heterostructures", Integrated Circuits Laboratory, Stanford University, 1994, p. 27.

119. Arora, N.D., Hauser, J.R., Roulston, D.J. "Electron and Hole Mobilities in Silicon as a function of Concentration and Temperature", IEEE Trans Electron Devices, ED-29,292-295 (1982).

120. Zhou, J.R. and Ferry, D.K., "Simulation of Ultra-small GaAs MESFET Using Quantum Moment Equations": IEEE Trans. Electron Devices, Vol. 39, Mar. 1992, pp 473-478

121. Zhou, J.R. and Ferry, D.K., "Simulation of Ultra-small GaAs MESFET Using Quantum Moment Equations - II:Volocity Overshoot", IEEE Trans. Electron Devices, Vol. 39, Aug. 1992, pp 1793-1796.

122. Yamada, T., Zhou, J.R., Miyata, H., and Ferry, D.K., "In-Plane Transport Properties of Si/ Si(1-x)Ge(x) Structure and its FET Performance by Computer Simulation", IEEE Trans. Electron Devices, Vol. 41, Sept. 1994, pp. 1513-1522.

123. Van Dort, M. J., Woerlee, P. H. and Walker, A. J. "A Simple Model for Quantisation Effects in Heavily-Doped Silicon MOSFETs at Inversion Conditions", Solid-State Elec., Vol. 37, No. 3, pp. 411-414, 1994.

124. Hansch, W., Vogelsang, Th., Kirchner, R., and Orlowski, M. "Carrier Transport Near the Si/ SiO2 Interface of a MOSFET", Solid-State Elec., Vol. 32, No. 10, pp. 839-849, 1989.

125. Hurkx, G.A.M, Klaassen, D.B.M., Knuvers, M.P.G., "A New Recombination Model for Device Simulation Including Tunneling," IEEE Trans. Electron Devices, Vol. ED-39, Feb. 1992, pp 331-338.

126. Simmons J.G. and Taylor G.W., Phys. Rev. B, 4, p. 502, 1971.

127. Chynoweth A.G., "Ionisation Rates for Electrons and Holes in Silicon", Phys. Rev., 109, pp. 1537-1540, 1958.

128. Baraff G.A., "Distribution Functions and Ionisation Rates for Hot Electrons in Semiconductors", Appl. Phys. Rev., 128, pp. 2507-2517, 1962.

129. Feigna C., Venturi F., “Simple and Efficient Modelling of EPROM Writing”, IEEE Trans. Elect. Dev., Vol. 38, pp. 603-610, 1991.

130. Cassi C. and Ricco B., “An Analytical Model of the Energy Distribution of Hot Electrons”, IEEE Trans. Elect. Dev., Vol. 37, pp. 1514-1521, 1990.

131. Wada, M. et al, “A Two-Dimensional Computer Simulation of Hot Carrier Effects in MOSFETs”, Proc. IEDM Tech. Dig., pp. 223-225, 1981.

132. Hall, R.N., “Electron Hole Recombination in Germanium”, Phys. Rev., 87, pp. 387, 1952.

133. Shockley W. and Read W.T., “Statistics of the Recombination of Holes and Electrons”, Phys. Rev., 87, pp. 835-842, 1952.

134. Beattie, A.R. and White, A.M., “An Analytical Approximation with a Wide Range of Applicability for Electron Initiated Auger Transitions in Narrow-gap Semiconductors”, J. Appl. Phys., Vol. 79, No. 12, pp. 802-813, 1996.

135. Tan, G-L., Bewtra, N., Lee, K. and Xu, J.M., “A Two-Dimensional Nonisothermal Finite Element Simulation of Laser Diodes”, IEEE Journal of Quantum Electronics, Vol. QE-29, No. 3, pp. 822-835, March 1993.

136. Lindefelt U., “Equations for Electrical and Electrothermal Simulation of Anisotropic Semiconductors”, J. Appl. Phys., 76, pp. 4164-4167, 1994.

137. Lades M. and Wchutka G., “Extended Anisotropic Mobility Model Applied to 4H/6H-SiC Devices”, Proc. IEEE SISPAD, pp.169-171, 1997.

138. Yang K, East J. and Haddad G., “Numerical Modeling of Abrupt Heterojunctions Using a Thermionic-Field Emission Boundary Condition”, Solid-State Electronics, Vol. 36, No. 3, , pp.321-330, 1993.

139. Vendelin G.D., “Design of Amplifiers and Oscillators by the S-parameter Method”, Wiley,New York, 1982, ISBN 0471092266.

140. Takahashi Y., Kunihiro K. and Ohno Y., “Two-Dimensional Cyclic Bias Device Simulator and Its Applications to GaAs HJFET Pulse Pattern Effect Analysis”, IEICE Journal C, June 1999.

[This page intentionally left blank]

Index

Symbols

"kink" 4-9

Numerics

3D 11-1, 13-1

A

Absorption 8-6
Absorption Loss 9-2, 9-5
AC Analysis 15-134, 15-138
Affinity 3-24, 5-3
AlAs 5-20
AlGaAs 5-20
ALIGN 2-21
Anti-Reflective Coatings 8-5, 15-42
A-stability 14-17
ATHENA....2-8
Auger coefficient 15-57
Auger Recombination.... 3-64, 15-97
Available Photocurrent 8-8
Avalanche breakdown 3-66

B

Bandgap Narrowing ....3-3, 3-7, 4-5, 4-8, 5-18, 5-23, ... 5-25,15-58, 15-98
Band-to-Band Tunneling 3-74, 15-97
Barrier Height 5-2
Barrier Lowering 3-25, 15-11
Batch Mode 2-2
BDF 2 15-73
Beam 15-5
Bias Stepping 15-137
BICGST 11-8, 14-21, 15-69
Bipolar 4-5, 6-1
Blaze3D 11-1
BLOCK 11-8, 14-4, 15-69
Block Methods 2-27, C-4
Boltzmann statistics 3-5
Boltzmann Carrier Statistics 15-98
Boundary Conditions 11-7
Breakdowm 2-37, 4-9, 15-15
Breakdown Voltage....2-36
Bulk Absorption Coeffiecient 15-59
Bulk Traps 3-11

C

Capacitance 12-1
Capture Cross Sections 3-13, 7-13
Carrier Heating 4-4, 4-9
Carrier-carrier Scattering 3-36
CCD 8-1, 8-9
C-Interpreter 2-26, 3-72, 8-7, A-1
Circuit Models 10-1
CLIM.DD 14-7
CLIM.EB 14-7
CLIMIT 2-36, 14-7, 15-71
Command Menu 2-7
Comments 15-2
Compliance 2-37, 15-133
Conductance 12-1
Conduction Band Density.... 15-56
Conductivity 12-1
Contact 2-17
Contact Resistance 2-18, 3-29
Continuation 15-2
Continuity Equations 3-1
Convergence Criteria 14-6, 14-10
Current Boundary 2-18, 2-36, 15-11
Current Boundary Conditions 3-27
Curve trace 2-37
Curve Tracing 15-131
Curve-tracer 14-16, 15-14
Cylindrical Coordinates 2-11

D

Dark Current 8-12
DeckBuild 2-2, 2-38
Defect 15-17
defect states 7-1
degeneracy 3-13
degradation 3-83
Density Of States 3-5, 3-20, 5-21, 5-23, 5-26
Detection Efficiency 8-13
DevEdit3D 11-3
Device Degradation 15-97
Device3D.... 11-1
dielectric constant 5-24, 5-26
dielectric permittivity 15-56
dipole lowering 15-11
Direct methods 14-5
Dirichlet 3-23
Discretization 14-1, 14-3
Displacement Current 3-4
Distributed Contact Resistance 15-12
Doping.... 2-11, 15-20

Drift Diffusion 5-12
Drift-diffusion Model 3-2

E

EEPROM 4-6
effective mass 5-22, 5-24
effective refractive index 9-4
Eigen Functions 9-1
Eigen Values 9-1
Einstein relationship 3-3
Electrode 2-11, 4-3, 4-5, 15-28
Electron Affinity 3-8, 5-21, 5-23, 5-26, 15-56
Electron Injection 15-97
ELIMINATE 2-10
Emission Rates 3-13
Energy Balance 2-22, 2-28, 3-4, 3-17, 3-19, 3-30, 3-56, 4-4, 4-6, 4-9, 5-17, 15-99
Energy Dependent Mobilities 3-22
Energy Gap 15-56
Energy Relaxation 3-21
EPROM 4-7
Error Norm 14-6
ESD 4-4, 6-1
Examples 2-4

F

Fast Fourier Transform 15-34
Fermi-Dirac 3-4, 5-13
Fermi-Dirac carrier statistics 3-5, 3-9, 3-10, 15-98
Ferroelectric 3-84
Floating Contacts 2-18, 3-26
Floating Field Plates 6-8
Floating Gates 4-6, 4-7, 15-11, 15-12
Floating Guard Rings 6-8
Fourier Analysis 14-19
Fowler-Nordheim tunneling 4-7, 15-97
Free-carrier 9-2
free-carrier loss 9-5
Freeze-out 3-9, 3-10

G

GaAs 5-20
gains 15-48
gate current 3-75, 4-7, 15-98
GDS-II 12-4
generation 5-18
Giga3D 11-1
GMRES 14-21, 15-69
GUMMEL 4-9, 11-8, 14-4, 15-69
Gummel method 2-27

H

Half-implicit Scheme 14-18
HBT 5-1, 5-24
Heat Capacitance 6-2
Heat Capacity 15-58
Heat Sink 6-2, 13-1
Heat-flow 4-4, 13-3
Helmholtz 9-1
HEMT 5-1, 5-24
Heterojunctions 2-21, 5-1, 15-16
Hole Injection 15-97
Hot Carrier Injection Mode 3-76
Hot Electron Injection 4-7
Hot Hole Injection 4-7

I

IGBT 6-1
ILUCGS 11-8, 14-21, 15-69
impact ionization 15-37, 15-98
Impact Ionization Models 3-66
incomplete ionization 3-9, 15-98
InGaAsP 5-22
Initial Guess 2-32, 14-15, 15-133
InP 5-22
Insulating contacts 3-27
Interactive Mode 2-2
Interface 2-21
Interface Charge 4-4
Interface Trapped Charge 3-11
Interface Traps ..... -11
Intrinsic Carrier Concentration 3-6, 3-7, 5-13
Inversion Layer Mobility 3-43
Ionization Integral 15-109, 15-135, 15-138
Iterative methods 14-5

K

Klaassen's Carrier Concentration
Dependent Model 3-64

L

Laplace's equation 12-1
LASER 9-1, 15-102
latch-up 2-37
Lattice Heating 2-27, 4-4, 4-9, 6-1
Leakage Current 4-9
LED 5-1, 8-16
Lifetimes 3-60, 3-61
Line Width Broadening Factor 15-60
LOCAL 14-16
Local Optical Gain 9-2
Log Files 2-1, 2-40, 15-47

low field mobility 3-31
L-stability 14-17
Lucky Electron 3-76
LUMINOUS 8-1
Luminous Efficiency 8-16
Luminous Intensity 8-16
Luminous Wavelength 8-16
Lumped Capacitance 15-12
Lumped Elements 3-28
Lumped Resistance 15-12

M

MaskViews 12-4
Material Parameters 2-17
Material Properties 2-20
Mathiesen's Rule 3-38, 3-44
Mesh 2-9, 4-8, 14-1, 15-64
METHOD Statement 2-29
mirror loss 9-5
MIXEDMODE 10-1, 11-2
MLOCAL 14-16
mobility 3-31, 4-5, 4-8, 5-15, 5-19, 5-24, 5-26, 15-57, 15-82, 15-96
Models 2-21
Monochromatic 8-10
MOS 4-1, 6-1
MSM 8-1, 8-9
Multispectral 8-10

N

negative differential mobility 5-16
negative differential resistance 4-9
Netlist 10-3
Neumann boundary 3-27
NEWTON 4-9, 11-8, 14-4, 15-69, C-4
Newton method 2-27
Newton-Richardson 14-4, 15-72
Non-volatile Memory 4-6
Numerical Methods 2-27, 15-67, C-3

O

Ohmic contacts 3-23
Open Circuit Voltage 4-6, 8-16
Optical Recombination 15-97
Optimizer 2-41

P

Parallel ATLAS 2-3
Parameter Extraction 2-40
Parameters 2-5, 15-1
Parasitic Element 15-49
Peltier coefficient 6-8
Photoconductor 8-1, 8-9
Photocurrent 8-8
Photodetectors 8-1, 8-9
Photodiodes 8-1, 8-9
Photogeneration 8-1, 8-6, 15-138, C-1
Photon Density 9-4
Photon Rate Equations 9-4
PISCES-II 2-7, 2-30
Poisson's Equation 3-1
Poisson-Schrodinger 15-98
Polarization 8-10, 15-6
Power Device 6-8
PREVIOUS 14-16
Probability Of Occupation 7-3
PROJECTION 14-16

Q

Quantum Efficiency 8-6, 8-8, 8-13, 15-16
Quantum Moments Model 15-98
Quantum3D 11-2
Quasi-Fermi 5-12
Quasistatic 15-73

R

Radiative (Direct) Recombination 3-64
Ray Trace 8-1
Recombination 4-8, 5-18
Reflection Coefficient 8-4
Refractive 8-11
Refractive Index 8-1, 9-2, 15-59
Regions 2-10
Regrid 2-13, 4-2
Reliability 3-83
Remesh 2-13, C-7
RHSNORM 14-8
Richardson Constant 3-25, 15-56
Run-Time Output 2-38

S

Saturation Velocity 3-56, 15-57
Scattering 3-31
Schottky 2-17, 3-24, 3-25, 5-3
Semi-implicit 15-70
Shockley-Read-Hall Recombination 3-60, 15-97
Short Circuit Current 8-16
SiGe 5-25
Silicon Carbide 5-27
Silvaco's Virtual Wafer Fab 2-1
Single Event Upset 11-5, 15-126
Small-signal 14-18
Small-Signal AC 2-34
Smoothing 2-14, 14-3, 15-65, 15-121

Snapback 2-37, 4-9
SOI 4-8
Solar Cells 8-1
Solution Files 2-1
Solution Tolerances 15-70
Source Photocurrent 8-8
Space Charge 3-11
S-parameter 15-48
Spectral Response 8-15
SPICE 10-1
SPISCES2 ..... B-1
SSUPREM3 2-13
Statements 2-5, 15-1
Stiffness 14-17
Stimulated Emission 9-4
Surface Recombination 3-65, 15-11, 15-41
Syntax 2-5, 15-1

T

Temperature Dependent Mobility 3-56
termination criteria 14-8
TFT 7-1
TFT3D 11-2
Thermal Boundary Conditions 6-6
Thermal Conductivity 6-2, 13-2, 15-59
Thermal Contact 11-7, 15-145
Thermal Resistance 6-6
Thermal Velocities 3-13
Thermionic Emission 5-14
Threshold Voltage 4-9
Tonyplot 15-147
Tonyplot3D 11-8
Transient Simulation 14-17, 15-137
Transient Solutions 2-35
Transient Traps 3-16, 7-4
Transmission Coefficient 8-4
Trap 2-33, 3-11, 15-149
Trap-Assisted Tunneling 3-14, 3-62, 7-5
TR-BDF 11-8, 14-18

U

UTMOST 15-151

V

valence band density 15-56
velocity saturation 4-5, 5-16, 5-26

W

wavelength 15-6
Workfunction 2-17, 3-24, 4-3, 5-2, 15-11, C-5

X

XNORM 14-8